SIGNET • 451-AE2598 • (CANADA $2.95) • U.S. $2.50

WHY SPEND MONEY ON EXPENSIVE SOFTWARE?
HERE ARE EASY-TO-PROGRAM, EXCITING-TO-PLAY
GAMES DESIGNED ESPECIALLY FOR YOUR
TIMEX SINCLAIR 1000™ AND 1500™

51 GAME PROGRAMS FOR THE TIMEX SINCLAIR 1000™ AND 1500™

BY TIM HARTNELL
AUTHOR OF
MASTERING YOUR TIMEX/SINCLAIR 1000™ PERSONAL COMPUTER

SIGNET
AE
2598

51 GAME PROGRAMS FOR THE TIMEX SINCLAIR 1000™ AND 1500™

TIM HARTNELL

U.S. $2.50
CAN. $2.95
0-451-12598-3
NONFICTION

51 GAME PROGRAMS FOR THE TIMEX SINCLAIR 1000(TM) AND 1500(TM)

by Tim Hartnell

A SIGNET BOOK
NEW AMERICAN LIBRARY
TIMES MIRROR

Published by arrangement with Addison-Wesley Publishing Company, Inc.

SIGNET TRADEMARK REG. U.S. PAT. OFF. AND FOREIGN COUNTRIES
REGISTERED TRADEMARK—MARCA REGISTRADA
HECHO EN CHICAGO, U.S.A.

SIGNET, SIGNET CLASSIC, MENTOR, PLUME, MERIDIAN AND NAL BOOKS are published by The New American Library, Inc., 1633 Broadway, New York, New York 10019

First Signet Printing, September, 1983

1 2 3 4 5 6 7 8 9

PRINTED IN THE UNITED STATES OF AMERICA

CONTENTS

DRIVING GAMES

BOARD GAMES AND SIMULATIONS

CARD GAMES

BRAIN GAMES

WORD AND LETTER GAMES

INTRODUCTION

With this book and your Timex Sinclair 1000™ or 1500™, you're set for a number of adventures. Despite their small size, the T/S 1000 and 1500 are computers of quite immense power, and this book contains 51 programs designed to show you just how great that power is, and how flexible your new computer can be. Note that for most of the games in this book you will need the Timex 16K expander.

The programs are divided into six sections:

- Moving graphics games
- Driving games
- Board games and simulations
- Card games
- Brain games
- Word and letter games

The section headings alone give some idea of the flexibility of the computer, and of the exciting range of programs in this book. We've got a wide range of games for you, from *Dragon's Gold*, *Breakout*, and *Galaxy Patrol*, to *Checkers Seven*, *Fastermind*, and *Tic Tac Toe*. The programs all contain ideas that you can adapt and use to enhance your own programs.

Games are great fun, and are one of the reasons that personal computers are as popular as they are today.

Moving Graphic Games

PROTECTOR

In this program, you are given the task of protecting a defective part of the force field guarding earth. A difficult task indeed. Certain aliens, who do not have kindly thoughts about dear old peace-loving (!) Earth, have also detected the weak spot in our defenses, and try to break through the field.

The field can stand one attack on any part, but this weakens that part. Any weak section which receives a second hit causes the collapse of the field and so leaves earth defenseless. The field will also collapse under the strain of having more than nine weak spots along its length.

Your task is to block the attacks with your craft. You can restore any weakened part of the field directly below you by pressing **F**. The **5** and **8** keys control your motion, moving you in the direction of the arrows on those keys.

There are five skill levels, with level 1 the easiest. Since the first few levels are really only for practice, the scoring system is biased toward the higher and harder levels. Failure, I'm afraid, is inevitable, because the aliens continually speed up their attacks if the preceding waves fail. Your score is given at the end of the game, along with the option of a second or subsequent

game. Pressing **N** at the end will end the game. *Protector* was written by Paul Toland.

```
   5 CLS
  10 LET D$≠"   ████████████████
███████"
  20 PRINT AT 19,3;"█";TAB 27;"█
"
  30 PRINT AT 0,10;"PROTECTOR"
  40 PRINT AT 20,5;"ENTER SKILL
LEVEL 1-5"
  50 IF INKEY$="" OR INKEY$<"0"
OR INKEY$>"5" THEN GOTO 50
  60 LET S=VAL INKEY$*2-1
  90 LET B=16
 100 LET V=1
 110 LET W=0
 120 FOR I=1 TO 20
 130 LET A=INT (RND*23)+4
 140 FOR H=S TO 19 STEP V
 145 PRINT AT 20,1;D$
 150 PRINT AT H,A;"T"
 160 PRINT AT 19,B;" "
 170 LET B=B+(INKEY$="8")-(INKEY
$="5")
 180 LET B=B+(B=3)-(B=27)
 190 PRINT AT 19,B;"0"
 200 IF INKEY$<>"F" OR D$(B)="█"
 THEN GOTO 230
 210 LET W=W-1
 220 LET D$(B)="█"
 230 PRINT AT H,A;" "
 240 NEXT H
 250 IF B=A THEN GOTO 290
 260 IF D$(A)="▒" OR W=9 THEN GO
TO 320
 270 LET W=W+1
 280 LET D$(A)="▒"
 290 NEXT I
 300 LET V=V*2
 310 GOTO 120
 320 PRINT AT 5,0;"YOU FAILED -
THE FORCE FIELD IS BROKEN AFTER
";((V-1)*20+I)*S;" WAS SCORED"
 330 PRINT ,,"DO YOU WANT TO TRY
 TO PROTECT    ANOTHER PLANET ?"
 340 IF INKEY$="" THEN GOTO 340
 350 IF INKEY$<>"N" THEN RUN
 360 STOP
```

GROUND-TO-AIR MISSILE

You have ten ground-to-air missiles (GAMs) under your command. Your job is to destroy the alien ships (which resemble Terran letter V's) before they land on Earth and destroy it. If they land, the game is over, and the number of ships you destroyed is shown in the top left-hand corner of the screen. The **1** key moves you left, **0** moves you right, and **2** moves you up the screen. You must get the + in front of the advancing V's to stop them. The screen clears after each successful hit, and at the end of the game. *Ground-to-Air Missile* was written by Aidan Walsh and Kevin McCarthy, Cork, Ireland.

```
 1 FOR N=PI-PI TO VAL "9"
 2 LET A=VAL "20"
 3 LET B=VAL "11"
 4 LET X=PI-PI
 5 LET Y=B
 6 FOR F=PI-PI TO VAL "63"
 7 PLOT F,PI-PI
 8 NEXT F
 9 PRINT AT A,B;CHR$ 21
10 PRINT AT X,Y;CHR$ 59
```

```
  11 PRINT AT A,B;CHR$ 0
  12 LET Z=INT (RND*VAL "3")
  13 LET X=X+1
  14 LET Y=Y+(Z=VAL "1")-(Z=VAL
"0")
  15 IF INKEY$="9" THEN LET A=A+
2
  16 LET B=B+(INKEY$="0")-(INKEY
$="1")
  17 LET A=A-(INKEY$="2")
  18 IF A=X AND B=Y OR X>=22 THE
N GOTO VAL "20"
  19 GOTO VAL "9"
  20 CLS
  21 IF A=X THEN NEXT N
  22 IF X=VAL "22" THEN PRINT N
```

BREAKOUT

In *Breakout*, you control the bat on the left-hand side of the screen using the **6** and **7** keys to move your bat in the direction shown by the arrows on the keys, to try and keep the ball in the "court." The aim is to demolish as much of three walls as possible. Walls farther back give higher scores. You are allowed five balls in each game, and a score over 2000 gives a new game.

The game is limited, to some extent, by the speed of the computer. You need to type in **LET H = 0** before you run the program to set the high score counter, then start the program by entering **GOTO 5**, rather than **RUN**. If you save the program after you've played it, then start running it with **GOTO 5**, the old high score will automatically be your target for the new game.

```
  5 RAND
 10 LET T=0
 15 LET Z=0
 17 LET Q=0
 20 LET TB=0
 25 LET Q=Q+1
 30 LET R=265
```

```
  40 PRINT AT 1,0;"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀BREAKOUT▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
  50 PRINT AT 17,0;"▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄BREAKOUT▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
  80 PRINT ,,"TOTAL 0     BALL NO
";
  90 PRINT ,,"  HIGHEST SCORE ";
H
 100 FOR I=2 TO 16
 110 PRINT AT I,13;"▚▚▚▚    ▌▌▌▀▀▀   ▌ "
 120 NEXT I
 130 LET A=-1
 140 LET D=1
 150 LET X=11
 160 LET Y=INT (RND*10)+5
 165 LET P=PEEK 16396+PEEK 16397
*256+1
 170 LET TB=TB+1
 180 GOTO (TB=6)*210+200
 200 POKE P+647,TB+28
 220 POKE R+P,0
 230 LET R=R+33*(INKEY$="6")-33*
(INKEY$="7")
 240 POKE P+R,5
 250 POKE P+Y*33+X,0
 260 LET X=X+A
 270 LET Y=Y+D
 275 LET Z=P+Y*33+X
 280 LET N=PEEK Z
 290 POKE Z,28
 300 IF Y=2 OR Y=16 THEN LET D=-
D
 310 IF N=5 OR X=30 THEN LET A=-
A
 320 POKE P+R,0
 330 LET R=R+33*(INKEY$="6")-33*
(INKEY$="7")
 340 POKE P+R,5
 350 GOTO (N<132)*30+360
 360 LET T=T+(136-N)*5
 370 PRINT AT 19,5;T
 380 LET A=-A
 390 GOTO (X>0)*90+130
 410 IF T>2000*0 THEN GOTO 20
 415 PRINT AT 10,3;"GAME OVER"
 420 IF T>H THEN LET H=T
 430 PRINT ,,"HIGH SCORE IS NOW
";H
 440 INPUT A$
```

```
450 IF A$="N" THEN STOP
460 CLS
470 GOTO 10
480 SAVE "BREAKOUT"
490 GOTO 10
```

ZOOMER

A long bar is printed on the screen, starting from a random position. A projectile then makes its way across the screen from the left. When you think the projectile is over the hole in the bar, press a key. If you're right, you'll get a point. You'll be pleased to see how your skill at this program improves as you continue. *Zoomer* was written by Nick Wilson.

```
 10 REM   ZOOMER
 11 REM NICK WILSON
 12 RAND
 20 LET M$=""
 30 FOR I=1 TO 32
 40 LET M$=M$+"█"
 50 NEXT I
 55 CLS
 60 LET L=INT (RND*21)
 70 PRINT AT L,0;M$
 80 LET K=INT (RND*31)
 90 PRINT AT L,K;" "
100 LET J=INT (RND*21)
105 IF J=L THEN GOTO 100
110 LET N=0
120 PRINT AT J,N;"▒"
130 LET V=N
140 LET N=N+1
150 PRINT AT J,V;" "
```

```
 155 IF N=30 THEN GOTO 55
 160 IF INKEY$="" THEN GOTO 120
 170 IF V=K OR V=K+1 OR V=K-1 TH
EN GOTO 190
 180 GOTO 120
 210 FOR I=1 TO 20
 220 NEXT I
 230 IF INKEY$<>"" THEN GOTO 120
 240 CLS
 250 PRINT AT 10,0;"TRY AGAIN  ?
"
 260 IF INKEY$="" THEN GOTO 260
 280 IF INKEY$="Y" THEN GOTO 55
 290 STOP
```

THUNDERBOLT

You are the gunner on a ground-station outpost, and it is your mission to stop spy planes from flying over you. The computer is your monitor (it is good to see how much the U.S. Government is willing to invest in protecting the country), and on the monitor you see the planes, and your ten missile launchers.

As the enemy planes fly over, you have to press the number on the missile that you want to fire. The missile then zips up the screen, either destroying the enemy or totally missing, depending on your skill.

There are two extra features which make it difficult. The plane keeps moving after you've fired, so you really have to fire in front of it to score a hit, and the missiles are not reloaded until you manage to bring down one of the enemy. You can change line 100 to any design you like, as long as there's a space at the start, and a graphic H in the middle. The game is simplified if you add more graphic H's to the design. *Thunderbolt* was written by Nick Wilson.

```
10 REM THUNDERBOLT
11 REM NICK WILSON
12 RAND
```

```
  20 LET B=0
  22 LET S=INT (RND*16)
  25 LET F=0
  26 CLS
  30 PRINT AT 21,0;
  40 FOR I=1 TO 32
  50 PRINT "█";
  60 NEXT I
  70 PRINT AT 20,20;"::::::::::"
  80 PRINT AT 21,20;"0123456789"
  90 PRINT AT S,B;
  95 LET B=0
  96 LET S=S+1
  97 IF S=21 THEN RUN
 100 PRINT AT S,B;" ▟▒▙"
 110 IF INKEY$<>"" THEN GOTO 170
 115 IF F=1 THEN GOTO 240
 130 LET B=B+1
 140 IF B=27 THEN GOTO 90
 150 GOTO 100
 170 LET A$=INKEY$
 175 IF CODE A$<28 OR CODE A$>37
 THEN GOTO 111
 190 LET R=20+VAL A$
 200 PRINT AT 20,R;
 210 IF PEEK (PEEK 16398+256*PEE
K 16399)<>14 THEN GOTO 111
 220 LET F=1
 225 LET P=20
 230 GOSUB 111
 240 LET P=P-1
 245 IF P<S-3 THEN GOTO 280
 250 PRINT AT P,R;"$";AT P+1,R;"
 ";
 252 PRINT AT P-1,R;
 253 IF PEEK (PEEK 16398+256*PEE
K 16399)=136 THEN GOTO 300
 270 GOTO 130
 280 LET F=0
 285 PRINT AT P+1,R;" "
 290 GOTO 111
 300 LET A$="▣*▣*▣"
 310 LET B$="*▣*▣*"
 320 FOR I=1 TO 6
 330 PRINT AT P-2,R-2;A$;AT P-2,
R-2;B$;AT P-1,R-2;A$;AT P-1,R-2;
B$;AT P,R-2;A$;AT P,R-2;B$
 340 NEXT I
 350 RUN
```

SURGE

In *Surge*, written by Tim Rogers, your ship (the $) is somewhere out near a strange asteroid belt. The asteroids are slabs. Your ship has a shield, so the asteroids cannot destroy your ship. The only problem is that you get pushed up the screen by any slabs you come in contact with. The aim of the game is to stay on the screen for as long as possible. The lower down you are, the more points you score.

The machine-code routine in the REM statement takes the place in line 80 of the BASIC line **IF PEEK (PEEK 16398 + 256 * PEEK 16399)**. There are seven characters after the **REM**, and in decimal they are 42, 14, 64, 78, 6, 0 and 201. All but CHR$ 78 can be entered from the keyboard; 78 has to be POKED, by line 10. You move your ship to the right by pressing any key, and it drifts left when you release the key.

```
 1 REM E:RND7▄▀ TAN
 5 LET H=1
10 POKE 16517,78
15 LET S=H-H
20 LET U=10
25 LET T=20
```

```
  30 LET P=U
  40 PRINT AT U,P;" "
  50 LET P=P-H/H*(P>H/H)
  60 LET P=P+(INKEY$<>"")*2*(P<T
)
  65 SCROLL
  70 PRINT AT U,P;
  80 IF USR 16514=CODE "▒" THEN
LET U=U-H/H
  90 IF U=H-H THEN GOTO 200
 100 PRINT AT U,P;"$"
 110 PRINT AT CODE ")",RND*T;"▒▒
▒▒▒"
 120 LET S=S+U
 130 GOTO 40
 200 IF H<S THEN LET H=S
 210 PRINT AT U,P;"SURGE",S,H
 220 PAUSE 4E4
 230 GOTO 15
```

PUSSY-GET

If you are a cat reading this, you're hereby given permission to change the program to *Human-Get.* Your job is to drop seven weights (printed across the top of the screen) onto six cats which run across the bottom of the screen, one after another. If you press the keys **2** to **7**, the corresponding weight will drop and the cat will stop running until the weight hits it, or sails on past. If the weight misses the cat, the frisky feline will flee. However, if it gets smashed by the weights, it will turn into a cross. Some heavy symbolism is in order here, as you can tell. The game can be made more difficult by removing lines 330 and 370, which speeds up the game, but in so doing, it removes the checks on human cheating. *Pussy-Get* comes from Nick Wilson.

```
 5 CLS
10 REM   PUSSYGET
11 REM NICK WILSON
12 DIM A(8)
13 LET CATS=6
19 LET B$="█"
20 FOR I=1 TO 5
30 LET B$=B$+B$
40 NEXT I
```

```
  50 PRINT B$
  60 FOR I=.7 TO 7
  70 PRINT TAB I*4;":"
  75 PRINT TAB I*4;"."
  80 PRINT TAB I*4;"█"
  90 PRINT TAB I*4;"█"
  95 PRINT AT 0,I*4;CHR$ (CODE (
STR$ (I))+129)
  96 LET A(I)=I*4
 100 PRINT AT 1,0;
 110 NEXT I
 140 LET T$=" ▌  ▐"
 150 LET U$=" ▙██▟"
 160 LET V$="  ██ "
 170 LET W$=" ▖  ▗"
 180 LET M=0
 185 PRINT AT 17,0;
 190 PRINT TAB M;T$
 200 PRINT TAB M;U$
 210 PRINT TAB M;V$
 220 PRINT TAB M;W$
 230 LET M=M+1
 240 IF M=27 THEN GOTO 270
 250 IF INKEY$="" THEN GOTO 185
 260 GOTO 320
 270 PRINT AT 17,0;
 280 FOR I=1 TO 4
 290 PRINT TAB M;"     "
 300 NEXT I
 310 GOTO 180
 320 LET A$=INKEY$
 330 IF A$<"1" OR A$>"7" THEN GO
TO 185
 340 LET A=VAL A$
 350 LET X=A(A)
 355 LET J=3
 360 PRINT AT J,X;
 370 IF PEEK (PEEK 16398+256*PEE
K 16399)<>128 THEN GOTO 185
 380 PRINT AT J,X;" ";AT J+1,X;"
█";AT J+2,X;"█"
 390 IF J+2=21 THEN GOTO 430
 400 PRINT AT J+3,X;
 410 IF PEEK (PEEK 16398+256*PEE
K 16399)<>0 THEN GOTO 450
 411 LET J=J+1
 420 GOTO 380
 430 LET J=0
 435 PRINT AT 21,X;" ";AT 20,X;"
:"
```

```
440 GOTO 185
450 PRINT AT 16,X;"  ";AT 15,X;"
"
459 PRINT AT 17,0;
460 PRINT TAB M-1;"   █"
470 PRINT TAB M-1;"  ███"
480 PRINT TAB M-1;"   █"
490 PRINT TAB M-1;"   █"
495 LET CATS=CATS-1
496 IF CATS=0 THEN GOTO 510
500 GOTO 180
510 CLS
520 PRINT AT 1,1;"TRY AGAIN ?"
530 IF INKEY$="" THEN GOTO 530
550 IF INKEY$="Y" THEN RUN
600 STOP
```

SNAKE

In this program, you must guide your snake, using the **5**, **6**, **7**, and **8** keys, toward the $'s in order to grow. You move in the direction of the arrows on those keys.

You must not hit the walls or yourself. The $'s are on the screen for a limited time only, so you must rush. The aim of the game is to make your snake grow as long as you possibly can. Even though this game is in BASIC, it is very fast, thanks to some clever string handling. The game was written by Paul Toland, whose best score to date is 55. Can you beat that?

At the end of a game, you get a new game by just pressing **ENTER**, and stop by pressing **N** then **ENTER**.

```
  5 CLS
 10 LET S$="██"
 20 LET L=2
 50 PRINT "████████████████████████████████"
 60 FOR I=1 TO 6
 70 PRINT "█";TAB 31;"█"
 80 NEXT I
 90 PRINT "████████████████████████████████"
```

```
  92 PRINT "SNAKESSNAKESSNAKESSN
AKESSNAKES"
  94 PRINT ,,"THE MONEY SNAKE ON
LY GROWS IF   FED WITH $$$.",,"Y
OU MUST GUIDE IT TOWARDS THE $ T
AKING CARE NOT TO HIT A WALL OR
ITSELF."
 100 LET M=L
 110 LET P=PEEK 16396+256*PEEK 1
6397+1
 120 LET M=INT (RND*30+1)+INT (R
ND*6+1)*33
 125 IF PEEK (M+P)>0 THEN GOTO 1
20
 130 POKE P+M,13
 140 FOR I=1 TO 30
 150 LET S=CODE S$(1)
 160 POKE P+CODE S$(L),0
 165 LET I$=INKEY$
 170 LET S=S+(I$="8")-(I$="5")+(
I$="6")*33-(I$="7")*33+(I$<"5" O
R I$>"8")*(S-CODE S$(2))
 175 LET N=PEEK (P+S)
 180 IF N<>13 AND N<>0 THEN GOTO
 270
 190 LET L=L+(N=13)
 200 POKE P+S,28
 210 LET S$=CHR$ S+S$(1 TO L-1)
 240 NEXT I
 250 IF PEEK (M+P)=13 THEN POKE
M+P,0
 260 GOTO 120
 270 PRINT "GAME UP--YOU MANAGED
 TO GROW TO","A LENGTH OF ";L,"T
RY AGAIN ?"
 280 INPUT A$
 290 IF A$<>"N" THEN RUN
```

PHARAOH'S REVENGE

This is an early Egyptian version of the "city bomb" type of program, in which you fly over a city, leveling skyscrapers in front of you with bombs dropped from your plane.

In this game, you're flying (on a magic carpet?) over a pyramid, and you have to try and destroy as *little* of it as possible, while aiming for a spy (a Phoenecian merchant who has not paid his transit taxes or spice import duty) hiding in the base of the pyramid. Press any key to drop a bomb onto the pryamid. You can make the game easier by adding **105 LET G = SIN PI**. *Pharaoh's Revenge* was written by Nick Wilson.

```
13 LET S=0
14 LET B=3
15 LET F=0
16 CLS
20 LET C=16
21 PRINT AT 10,0;
25 LET A$="███████████████████
████"
30 FOR I=1 TO 23 STEP 2
40 LET C=C-1
50 PRINT TAB C;A$( TO I)
60 NEXT I
```

```
  70 PRINT AT 21,15;"O"
  75 PRINT AT S,B;"      "
  80 IF RND>.94 THEN LET S=S+1
  90 LET B=3
 100 PRINT AT S,B;" ███▄ "
 110 LET B=B+1
 120 IF B=27 THEN GOTO 75
 125 IF F=1 THEN GOTO 200
 130 IF INKEY$="" THEN GOTO 100
 140 LET F=1
 145 LET J=S+1
 160 LET D=B+2
 170 GOTO 100
 200 PRINT AT J-1,D;" ";AT J,D;"
"
 201 LET J=J+2
 205 IF J>=22 THEN GOTO 260
 210 PRINT AT J,D;
 220 IF PEEK (PEEK 16398+256*PEE
K 16399)=52 THEN RUN
 230 PRINT "*"
 250 GOTO 100
 260 LET F=0
 270 GOTO 100
```

DROPPER

In this intriguing game from Nick Wilson, you have to fill a glass of water with stones as quickly as possible.

You'll see the glass, full of water, printed on the screen when you press **RUN**. There is a barrier along the top of the screen, and the stones move along the bottom of it.

When you think a stone is above the glass, press any key, and the stone will start to fall. If it lands outside the glass, another stone will appear, so you can have another try. If, however, the stone falls into the glass, it will fall to the bottom, or rest on top of another stone. When you've filled two complete rows along the bottom, the game will be over, and you'll be told how many stones it took you to fill it. If you'd like to fill three rows before the game ends, change line 240 to: **IF K = 19 OR K = 18 OR K = 17 THEN LET S = S + 1**.

```
10 REM    DROPPER 1
11 REM NICK WILSON
12 LET F=0
13 LET K=1
14 LET S=1
```

```
  15 LET P=0
  16 CLS
  20 FOR I=1 TO 31
  30 PRINT "█";
  40 NEXT I
  70 FOR I=11 TO 19
  80 PRINT AT I,12;"█▒▒▒▒█"
  90 NEXT I
 100 PRINT TAB 12;"▙████▟"
 110 LET X=1
 120 PRINT AT 1,X;"█"
 130 LET X=X+1
 135 PRINT AT 1,X-1;" "
 140 IF X=31 THEN GOTO 280
 150 IF INKEY$="" THEN GOTO 120
 160 LET M=X
 190 LET K=K+1
 195 IF K=21 THEN GOTO 260
 200 PRINT AT K,M;
 210 LET T=PEEK (PEEK 16398+256*
PEEK 16399)
 215 IF T=128 THEN GOTO 240
 220 PRINT "█";AT K-(PI-PI),M;CH
R$ T
 230 GOTO 190
 240 IF K=19 OR K=18 THEN LET S=
S+1
 255 PRINT AT K-1,M;"█"
 256 IF S=8 THEN GOTO 311
 260 LET K=0
 265 PRINT AT 0,0;8-S
 270 GOTO 150
 280 LET X=1
 290 LET P=P+1
 300 IF P=11 THEN GOTO 320
 310 GOTO 150
 311 PRINT AT 0,0;"WELL DONE ...
YOU TOOK ";P;" PASSES"
 312 GOTO 330
 320 PRINT AT 0,0;"RAN OUT OF TI
ME...SCORED ";S
 330 PRINT AT 4,0;"TRY AGAIN (Y
OR N) ?"
 340 IF INKEY$="" THEN GOTO 340
 360 IF INKEY$="Y" THEN RUN
```

SPACE DOCKER

From deepest space comes *Space Docker*, which simulates the docking of two spaceships. You'll see the hulls of the two ships on the screen, with your ship on the right, and an enemy ship on the left. You line your docking tube up with that of the other ship, using the **2** and **Z** keys, and pressing **P** when you think you've docked. This sounds simple, but there is a catch (as usual). The enemy's docking tube (an inverse space, just like yours) is moving in random up and down steps in an attempt to stall you, so you need to be reasonably quick on the keyboard finger to change directions as the other docking tube is doing. There's also a time limit working against you. If you don't dock quickly enough, the enemy ship will blow up, and take your ship with it.

At the end of the game, you'll receive a rating, depending on how well you did. All the graphics in the program are from the **H** key.

Nick Wilson wrote *Space Docker*.

```
10 REM SPACE-DOCKER
11 REM NICK  WILSON
12 LET S=0
```

```
  13 LET K=0
  30 FOR I=1 TO 21
  40 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒  ▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒"
  50 NEXT I
  51 PRINT AT 0,1;S
  60 LET DS=INT (RND*20)+1
  70 LET SD=INT (RND*20)+1
  79 FOR K=1 TO 200
  80 PRINT AT DS,14;"█";AT DS-1,
14;" ";AT DS+1,14;" "
  90 PRINT AT SD,15;"█";AT SD-1,
15;" ";AT SD+1,15;" "
  95 IF SD=DS AND INKEY$="P" THE
N GOTO 200
 100 IF INKEY$="2" AND SD>0 THEN
 LET SD=SD-1
 110 IF INKEY$="Z" AND SD<19 THE
N LET SD=SD+1
 120 LET L=RND
 130 IF L>.5 THEN LET P=1
 140 IF L<.5 THEN LET P=-1
 145 IF DS+P>20 OR DS+P<0 THEN G
OTO 170
 146 LET DS=DS+P
 170 NEXT K
 172 LET S=S-200
 173 PRINT AT DS,14;"  "
 174 GOTO 51
 200 FOR I=1 TO 6
 210 PRINT AT DS,12;"DOCKED"
 215 LET L=SIN PI
 220 PRINT AT DS,12;"▒▒  ▒▒"
 225 LET L=SIN PI
 230 NEXT I
 240 LET S=S+(200-K)
 250 GOTO 51
```

GALAXY PATROL

Galaxy Patrol places you in command of a galactic patrol ship, which bears an uncanny resemblance to the letter V.

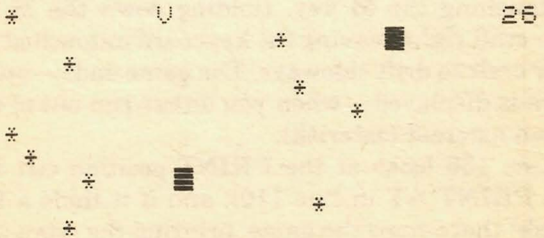
Your V-wing fighter starts with 50 gallons of fuel, which slowly decreases. The amount of fuel left is shown in the top right of the screen (in the screen printout the fuel figure is 33).

You refuel your fighter by hitting any of the random fuel dumps (inverse spaces). Each time you run through a fuel dump, you get 25 gallons. You control your craft by touching the **M** key. Holding down the **M** moves your craft right; leaving the keyboard untouched allows your craft to drift sideways. The game ends—and your score is displayed—when you either run out of fuel or hit an asteroid (asterisk).

Line 130 looks at the **PRINT** position (set by the last **PRINT AT** in line 110), and if it finds a **23** (asterisk) there stops the game, printing the score (**S**) and using an unassigned variable (**D**) to halt the game. If it finds a **128**, the computer knows you are running into a fuel dump, so the fuel is incremented by 25 (**LET F = F + 25).** Line 145 stops the game if you've run out of fuel (that is, if **F = Ø**).

Galaxy Patrol was written by R. Stubbs, based on a program by Tim Hartnell.

```
  5 LET F=50
 10 LET S=F-F
 20 LET A=S
 30 LET B=13
 40 LET C=10
 50 PRINT AT C,RND*30;"*"
 55 LET R=INT (RND*10)+1
 57 IF R=9 THEN PRINT AT 8,RND*30;"█"
 70 LET S=S+1
 75 LET F=F-1
 80 SCROLL
 90 IF B>2 THEN LET B=B-1
100 IF INKEY$="M" AND B<28 THEN LET B=B+2
110 PRINT AT A,B;"V";TAB 29;F;AT 1,B;
130 LET M=PEEK (PEEK 16398+256*PEEK 16399)
135 IF M=23 THEN PRINT S
140 IF M=128 THEN LET F=F+25
145 IF F=0 THEN PRINT S
150 GOTO 50
```

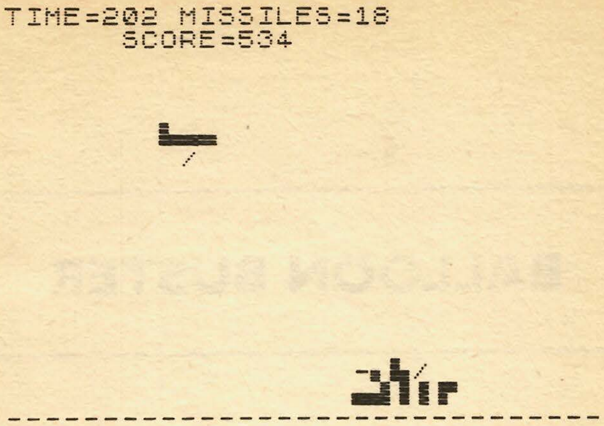


SEA RAIDER

This program is from Martin Frobisher, and it is more difficult to play than might be thought at first. In *Sea Raider*, you have to try to destroy a battleship by bombing it with your plane as you fly over. To make this more difficult, you fly twice as fast as the ship, and from time to time are buffeted by winds, increasing your speed even more.

You have 20 bombs in this version of the game, but you can easily change this by changing the value assigned to **M** in line 10. You fire by pressing **F**. You do not see anything fall from the plane, but if you hit it, you are rewarded by a rather odd explosion on the ship, which is immediately, miraculously, restored and continues on its tireless trip from left to right. There is a time limit of 300 seconds, and this is reduced steadily while the game is progressing, although you only see a new "reduced" figure every so often. The game ends when you run out of bombs or out of time. The maximum possible score is 5340, but it is practically impossible to get this within the time.

```
   1 REM      SEA RAIDER
   2 REM          BY
   3 REM    MARTIN FROBISHER
   4 REM
   5 PRINT AT 17,0;"-----------
-------------------"
  10 LET M=20
  15 LET T=300
  20 LET S=0
  30 LET A=1
  40 LET B=INT (RND*15)+1
  90 LET A=A+2
 100 LET B=B+1
 110 IF RND>.7 THEN LET A=A+1
 115 IF A>26 THEN PRINT AT 5,A-3
;"    ";AT 6,A-3;"       "
 120 IF A>26 THEN LET A=3
 125 IF B>26 THEN PRINT AT 15,B-
1;"      ";AT 16,B-1;"       "
 130 IF B>26 THEN LET B=1
 140 PRINT AT 5,A-3;"  ▄▄▄";AT
6,A-2;"  /"
 150 LET T=T-1
 155 IF T<1 THEN GOTO 205
 160 PRINT AT 15,B-1;" -▪▙/";AT
16,B-1;" ▄▟█▛"
 180 IF INKEY$="F" THEN GOTO 200
 185 LET T=T-1
 187 IF T<1 THEN GOTO 205
 190 GOTO 50
 200 LET M=M-1
 205 PRINT AT 0,0;"TIME=";T;" MI
SSILES=";M;"  "
 207 IF M=0 OR T<1 THEN STOP
 210 IF ABS (A-B)>2 THEN GOTO 50
 220 FOR Q=1 TO 5
 230 PRINT AT 15,B;"█ █ █";AT 15
,B;" ▒ ▒ ";AT 16,B;"▗▞▚▞▖";AT 16
,B;"▚▗▞▚"
 235 NEXT Q
 240 LET S=S+267
 250 PRINT AT 1,6;"SCORE=";S;" "
 265 IF INKEY$<>"" THEN GOTO 265
 270 GOTO 140
```


TIME=202 MISSILES=18
SCORE=534

BALLOON BUSTER

Moving left and right at the bottom of the screen, you fire upward at a row of balloons, floating against the ceiling. You have to try to burst them all. You move using the **0** key for right, the **1** key for left.

This sounds pretty simple, until you discover that if an empty space is fired into, another balloon appears, thus ensuring it will take longer to burst them all. You have no control over your firing rate, which complicates matters even further. To alter the frequency of fire, change the **.91** in line 125.

You can change the number of balloons at the start of the game, by altering the **9** in lines 13 and 50. The balloons can be changed to any character you like, providing that you change the **52** in line 190 to the code of the character chosen. Nick Wilson is the author of *Balloon Buster*.

```
10 REM      BUSTER
11 REM NICK WILSON
12 RAND
13 LET K=9
14 CLS
20 FOR I=1 TO 32
```

```
  30 PRINT "▒";
  40 NEXT I
  50 FOR I=1 TO 9
  60 PRINT AT 1,RND*29+1;
  70 IF PEEK (PEEK 16398+256*PEE
K 16399)<>0 THEN GOTO 60
  80 PRINT "O"
  90 NEXT I
  95 LET L=INT (RND*28)
 100 PRINT AT 21,L;" █ "
 110 IF INKEY$="0" AND L<28 THEN
 LET L=L+1
 120 IF INKEY$="1" AND L>0 THEN
LET L=L-1
 125 IF RND>.91 THEN GOTO 140
 130 GOTO 100
 140 FOR I=21 TO 2 STEP -1
 150 PRINT AT I,L+1;"$"
 160 PRINT AT I,L+1;" "
 170 NEXT I
 180 PRINT AT 1,L+1;
 190 IF PEEK (PEEK 16398+256*PEE
K 16399)=52 THEN GOTO 220
 210 PRINT "O"
 215 LET K=K+1
 216 GOTO 240
 220 PRINT " "
 230 LET K=K-1
 240 PRINT AT 0,0;K;" "
 245 IF K=0 THEN GOTO 260
 250 GOTO 100
 260 CLS
 270 PRINT AT 10,10;"TRY AGAIN ?
"
 280 IF INKEY$="" THEN GOTO 280
 300 IF INKEY$="Y" THEN RUN
```

I LOVE THE SOUND OF BREAKING GLASS

The object of this game is to protect a plate-glass window from projectiles which are being hurled at it. You do this by moving yourself up or down, by pressing the **2** or **Z** keys. If the ball hits you, it vanishes, and another one appears in its place. If you let the ball past, it will smash the window and your score will be given. You can make the game simpler by adding a few delay loops to slow it down. *I.L.T.S.O.B.G.* was written by Nick Wilson.

```
   15 LET TM=0
   20 CLS
   22 FOR I=1 TO 21
   23 PRINT TAB 15;"▒▒▒▒▒▒▒▒▒▒▒▒"
   24 NEXT I
   30 LET SX=INT (RND*21)+1
   40 LET SY=0
   50 LET HX=13
   60 LET HY=INT (RND*20)+1
   65 PRINT AT 0,0;TM
   70 PRINT AT HY,HX;" ";AT HY+1,
HX;"█ ";AT HY+2,HX;" "
   80 IF INKEY$="0" AND HY<19 THE
N LET HY=HY+1
```

```
 90 IF INKEY$="1" AND HY>0 THEN
 LET HY=HY-1
 100 PRINT AT SX,SY;
 102 IF PEEK (PEEK 16398+256*PEE
K 16399)=CODE "▌" THEN GOTO 150
 103 PRINT "▒";AT SX,SY-1;" "
 105 IF SY=15 THEN GOTO 130
 110 LET SY=SY+1
 120 GOTO 70
 130 PRINT AT SX,SY;"** SMASH **
"
 135 PRINT
 136 PRINT TAB 15;" YOU SAVED "
 138 PRINT TAB 15;" THE WINDOW"
 139 PRINT TAB 15;TM;"  TIMES
"
 140 STOP
 150 PRINT AT SX,SY-1;" "
 151 PRINT AT HY+1,HX;" "
 152 LET TM=TM+1
 160 GOTO 30
```

LETTER CHASER

Let your Ø run around the screen, to "run over" the letters you see in alphabetical order. You enter your speed setting for the game (from 1 to 5) and then use the **5** and **8** keys to move in the direction indicated by the arrows on those keys. The game ends if you hit a letter out of sequence. Pressing **ENTER** at the end will give you a new game, while **N** then **ENTER** will stop the program. You'll find yourself returning to this game, time and again, trying to complete it successfully. *Letter Chaser* was written by Paul Toland.

```
   8 PRINT "       LETTER CHASER
    "
   9 PRINT "ENTER SPEED SETTING
","1( FAST ) TO 5( SLOW )"
  10 LET X=1
  20 LET Y=X
  30 LET A=X
  40 LET D=0
  50 LET NC=38
  55 INPUT S
  57 CLS
  60 LET P=PEEK 16396+PEEK 16397
*256+1
  70 PRINT "         LETTER CHASE
R           "
```

```
 80 FOR I=1 TO 20
 90 PRINT "█";TAB 31;"█"
100 NEXT I
110 PRINT "           LETTER CHASE
R             "
120 FOR I=38 TO 63
130 LET RX=INT (RND*30)+1
140 LET RY=INT (RND*19)+2
150 IF PEEK (P+RX+RY*33)>0 THEN
GOTO 140
160 PRINT AT RY,RX;CHR$ I
170 NEXT I
180 LET I$=INKEY$
185 LET A=(I$="8")-(I$="5")+(I$
="")*A
190 LET D=(I$="6")-(I$="7")+(I$
="")*D
200 PRINT AT Y,X;" "
210 LET X=X+A
220 LET Y=Y+D
230 LET N=PEEK (P+Y*33+X)
240 PRINT AT Y,X;"0"
242 FOR J=1 TO 5
244 NEXT J
250 IF N=0 THEN GOTO 180
260 IF N<>NC THEN GOTO 310
270 LET NC=NC+1
280 IF NC<64 THEN GOTO 180
290 PRINT AT 10,10;"YOU MADE IT
????"
300 GOTO 320
310 PRINT AT 10,10;CHR$ (NC+128
);"-HARD LUCK"
320 PRINT AT 12,10;"TRY AGAIN??
"
330 INPUT A$
335 CLS
340 IF A$<>"N" THEN RUN
```

JET FIGHTER

You are the pilot of a defender jet. You must line up the enemy plane (a Ø) in your sights, using the **5**, **6**, **7**, and **8** keys, moving in the direction of the arrows on those keys. You destroy it by pressing the **F** key.

However, the enemy jet does not just sit there waiting for you to destroy it. You have to cope with its somewhat random evasive movements. Since you can only control your own plane, the enemy plane will appear to move in the direction opposite to the one you press.

The game, as listed, is at the beginner's level. The speed is dramatically increased if you remove line 16Ø, the time display. You press **ENTER** at the end for a new game, or **N** then **ENTER** to stop. *Jet Fighter* was written by Paul Toland.

```
  10 LET T=0
  20 LET X=INT (RND*32)
  30 LET Y=INT (RND*22)
  40 PRINT AT 10,14;"> <"
  50 IF T<>INT (T/2)*2 THEN GOTO
 80
  55 IF X>-1 AND X<32 AND Y>-1 A
ND Y<22 THEN PRINT AT Y,X;" "
```

```
  60 LET Y=Y+INT (RND*3)-1
  70 LET X=X+INT (RND*3)-1
  80 LET A$=INKEY$
  85 IF X>-1 AND X<32 AND Y>-1 A
ND Y<22 THEN PRINT AT Y,X;" "
  90 LET X=X+1*(A$="5")-1*(A$="8
")
 100 LET Y=Y+1*(A$="7")-1*(A$="6
")
 130 IF X>-1 AND X<32 AND Y>-1 A
ND Y<22 THEN PRINT AT Y,X;"0"
 140 IF A$="F" AND X=15 AND Y=10
 THEN GOTO 180
 150 LET T=T+1
 160 PRINT AT 0,0;T
 170 GOTO 40
 180 PRINT AT 10,15;"X"
 190 PRINT AT 20,0;"YOU GOT IT I
N ";T;" SECS"
 200 PRINT "TRY AGAIN?"
 210 INPUT A$
 220 IF A$="N" THEN STOP
 230 CLS
 240 GOTO 10
```

ZAP

You are trying to prevent the ubiquitous aliens from landing (the story of our lives).

They can descend in any one of three directions, straight down or diagonally down from the right or left.

You must position your craft using the **5**, **6**, **7**, and **8** keys (moving in the direction of the arrows on those keys), and fire your missile using **F** so that the missile intersects the alien's descent path.

Blocking the alien with your craft will have no effect. At the start of each game, you are asked for a difficulty level (Ø to 5), with Ø as the easiest level. The aim of this game, as you have probably guessed, is to prevent a landing for as long as you can. You press **ENTER** to get a new game, or **N** then **ENTER** to stop. *Zap* was written by Paul Toland.

```
 5 RAND
10 LET T=0
12 PRINT "LEVEL ?(0-5)"
14 INPUT L
15 LET BP=999
20 LET G=16
30 LET B=-1
35 CLS
```

```
  37 PRINT AT 19,0;"▄█▀▀▙▝▄▄▀▀▀▀▀▄▀▄▄▄▄▄▄▀▀▀▄▄▄▄▄▄▀▀▄█▛▙"
  40 LET T=T+1
  50 LET R=INT (RND*3)+1
  60 PRINT AT 0,0;T
  70 IF R=3 THEN LET P=INT (RND*
12)
  80 IF R=2 THEN LET P=INT (RND*
16)*2
  90 IF R=1 THEN LET P=INT (RND*
12)+20
 100 LET I=R-2
 110 FOR J=L TO 19
 120 PRINT AT J,P;"▒"
 130 LET A$=INKEY$
 140 PRINT AT 18,G;" "
 150 LET G=G+(A$="8")*2-(A$="5")
*2
 170 PRINT AT 18,G;"█"
 180 IF A$="F" AND B=-1 THEN LET
 B=17
 190 IF B=17 THEN LET BP=G
 200 IF B>-1 THEN PRINT AT B,BP;
" "
 210 IF B>-1 THEN LET B=B-1
 220 IF B>-1 THEN PRINT AT B,BP;
"*"
 230 IF (B=J OR B=J+1) AND BP=P
THEN GOTO 30
 240 PRINT AT J,P;" "
 250 LET P=P+I
 260 NEXT J
 270 PRINT AT J-1,P;"▒"
 280 PRINT "▒ ▒ THE ALIENS HAVE
LANDED ▒ ▒"
 290 PRINT "TRY AGAIN?  ";
 300 INPUT A$
 310 IF A$="N" THEN STOP
 330 GOTO 10
```

AVOID

Direct your ever-growing snake, using **5**, **6**, **7**, and **8**, so that it avoids the surrounding box, its own trail, and the + signs. It is allowed to hit five of the pluses before the game ends. Since each move decreases the space available, it is advisable to develop some movement tactics. The object of the game is to last as long as possible, and your time is given at the end of the game. Press **ENTER** for a new game, or **N** then **ENTER** to stop. *Avoid* was written by Paul Toland.

```
  10 RAND
  20 FOR I=0 TO 31
  30 PRINT AT 0,I;"█"
  40 PRINT AT 21,I;"█"
  50 NEXT I
  60 FOR I=0 TO 21
  70 PRINT AT I,0;"█"
  80 PRINT AT I,31;"█"
  90 PRINT AT RND*19+1,RND*29+1;
"+"
 100 NEXT I
 110 LET T=0
 115 LET H=0
 120 LET D=1
 125 LET X=2
 130 LET A=0
```

```
 135 LET Y=5
 140 LET P=PEEK 16396+PEEK 16397
*256+1
 150 LET A$=INKEY$
 170 IF A$="5" OR A$="8" THEN LE
T D=0
 180 IF A$="5" OR A$="8" THEN LE
T A=SGN (VAL A$-6)
 190 IF A$="6" OR A$="7" THEN LE
T A=0
 200 IF A$="6" OR A$="7" THEN LE
T D=SGN (VAL A$-6.5)*-1
 210 IF A$="7" THEN LET D=-1
 212 LET X=X+A
 214 LET Y=Y+D
 220 LET N=PEEK (P+33*Y+X)
 225 POKE P+33*Y+X,128
 230 IF N=21 THEN LET H=H+1
 250 IF N=128 OR H=6 THEN GOTO 2
90
 260 LET T=T+1
 280 GOTO 150
 290 PRINT AT Y,X;"*"
 295 PRINT AT 10,6;"YOU LASTED "
;T;" SECS."
 300 PRINT "NEW GAME ?"
 310 INPUT A$
 320 IF A$="N" THEN STOP
 330 CLS
 340 RUN
```

CENTROPOID

This is similar to Avoid only much more frantic. You (the checkered block) travel around the screen, hitting the asterisk while avoiding the solid blocks. You must hit all 10 of the asterisks before the game ends. When (or if) you're successful, your time is given.

Again motion is controlled by **5**, **6**, **7**, and **8**, and you get a new game by pressing **ENTER**, and stop by entering **N** then **ENTER**. *Centropoid* was written by Paul Toland.

```
 10 RAND
 20 FOR I=0 TO 31
 30 PRINT AT 21,I;"█"
 35 PRINT AT 0,I;"█"
 40 NEXT I
 50 FOR I=0 TO 21
 60 PRINT AT I,0;"█"
 70 PRINT AT I,31;"█"
 75 PRINT AT RND*19+1,RND*29+1;
"█"
 80 NEXT I
 90 FOR I=1 TO 10
100 PRINT AT RND*19+1,RND*29+1;
"*"
110 NEXT I
120 LET T=0
```

```
 130 LET H=0
 140 LET X=3
 150 LET Y=3
 160 LET A=0
 170 LET D=1
 180 LET P=PEEK 16396+PEEK 16397
*256+1
 190 LET A$=INKEY$
 195 POKE (P+33*Y+X),0
 200 IF A$="5" OR A$="8" THEN LE
T D=0
 210 IF A$="5" OR A$="8" THEN LE
T A=SGN (VAL A$-6)
 220 IF A$="6" OR A$="7" THEN LE
T A=0
 230 IF A$="6" OR A$="7" THEN LE
T D=SGN (VAL A$-6.5)*-1
 240 LET X=X+A
 250 LET Y=Y+D
 260 LET N=PEEK (P+33*Y+X)
 270 POKE (P+33*Y+X),6
 280 IF N=23 THEN LET H=H+1
 290 IF H=10 THEN GOTO 320
 300 IF N=128 THEN GOTO 330
 305 LET T=T+.5
 310 GOTO 190
 320 PRINT AT 10,8;"FINISHED IN
";INT T;" SECS."
 330 PRINT AT 11,10;"ANOTHER GAM
E ?"
 340 INPUT A$
 350 IF A$="N" THEN STOP
 360 CLS
 370 RUN
```

TOAD IN THE HOLE

Your task is to steer a "toad" (reincarnated, it appears, as an inverse asterisk) into its hole. You'll see the ground and hole near the bottom of the screen, and the toad will begin a rapid descent from the top. The 0 key moves your toad right, the **1** key moves it left. Once the toad reaches ground level, and depending on whether or not you got it home, you'll be given a score. You can alter the skill level by changing the size of the hole, playing around with the values in line 80. The graphic character in line 90 is from the **H** key. Lines 15 to 100 print the ground and hole, the routine from 110 to 205 moves the toad, and lines 300 to 340 print out the score and final message. *Toad in the Hole* was written by Nick Wilson.

```
10 REM TOAD-IN-THE-HOLE
11 REM NICK WILSON 1982
15 CLS
20 LET M=INT (RND*3)+18
30 PRINT AT M,0;
40 FOR I=1 TO 32
45 PRINT "▆";
50 NEXT I
70 PRINT AT M,INT (RND*22)+5;
```

```
  80 FOR I=1 TO INT (RND*3)+2
  90 PRINT "▒";
 100 NEXT I
 110 LET X=-1
 120 LET Y=INT (M-(RND*20)+(RND*
20))
 121 IF Y<0 OR Y>31 THEN GOTO 12
0
 125 LET S=1
 126 LET R=INT (53+RND*6)
 130 LET X=X+1
 150 PRINT AT X+1,Y;
 155 LET S=S+R
 160 LET T=PEEK (PEEK 16398+256*
PEEK 16399)
 170 IF T=128 THEN GOTO 300
 175 IF T=136 THEN GOTO 310
 180 PRINT "█";
 190 IF INKEY$="0" THEN IF Y<31
THEN LET Y=Y+1
 200 IF INKEY$="1" THEN IF Y>-1
THEN LET Y=Y-1
 205 GOTO 130
 300 PRINT S;AT 0,0;"CRASHED"
 305 GOTO 320
 310 PRINT S;AT 0,0;"IN THE HOLE
"
 320 FOR I=1 TO 80
 330 NEXT I
 340 RUN
```

MINEFIELD

In this game you are commander of a tank. Before you lies an enemy minefield through which you must pass. Can you do it? The allied forces are counting on you.

When you run the program the computer will generate a minefield—a different one each time you play. Drive your tank across the screen, using keys **5** through **8**.

If you hit a mine, your tank blows up and you lose. Pass safely through the minefield and you've won.

To input the program, first enter lines 1 through 45 as shown below:

```
   1 REM 12345678901234567890
   5 LET W$="2A0C4006172B237EFE7
6200310F8C9C6807718F2"
   6 LET N=1
  10 FOR X=16514 TO 16533
  20 LET K$=W$(N TO N+1)
  30 POKE X,16*CODE K$+CODE K$(2
)-476
  40 LET N=N+2
  45 NEXT X
```

Next, run this short subroutine. After running it, list the program and examine line 1. It should appear as follows:

```
    1 REM E£RND▞*F7 SAVE TAN LEN
 ▆?/ PAUSE
```

Now delete lines 5 through 45 and enter the remainder of the program, (starting at line 2).

```
    1 REM E£RND▞*F7 SAVE TAN LEN
 ▆?/ PAUSE
    2 REM ***MINEFIELD***
   50 CLS
   80 LET T=0
   90 LET S=0
  100 LET A=RND*40+40
  110 FOR I=1 TO A
  120 PRINT AT INT ((RND*19)+2),I
NT ((RND*29)+2);"▒"
  130 NEXT I
  140 LET P=INT (RND*19)+2
  150 LET O=0
  160 PRINT AT P,O;"I"
  170 LET Y=P
  180 LET X=O
  190 LET M$=INKEY$
  200 LET T=(M$="6")-(M$="7")+(M$
="")*T
  210 LET S=(M$="8")-(M$="5")+(M$
="")*S
  220 IF X+S<0 OR X+S>31 OR Y+T<2
 OR Y+T>21 THEN GOTO 170
  230 PRINT AT Y+T,X+S;
  235 IF PEEK (PEEK 16398+256*PEE
K 16399)=178 THEN GOTO 300
  236 PRINT "I"
  240 PRINT AT P,O;" "
  250 LET P=Y+T
  260 LET O=X+S
  265 IF O=31 THEN GOTO 400
  270 GOTO 170
  300 FOR B=1 TO 16
  310 RAND USR 16514
  320 NEXT B
  330 CLS
```

```
340 PRINT AT 10,12;"YOU LOSE"
350 GOTO 410
400 PRINT AT 0,12;"YOU WIN"
410 FOR L=1 TO 50
420 NEXT L
430 CLS
440 PRINT AT 10,10;"PLAY AGAIN
?","                  (Y/N)"
450 IF INKEY$="" THEN GOTO 450
460 IF INKEY$="Y" THEN RUN 50
470 STOP
```

DUCK SHOOT

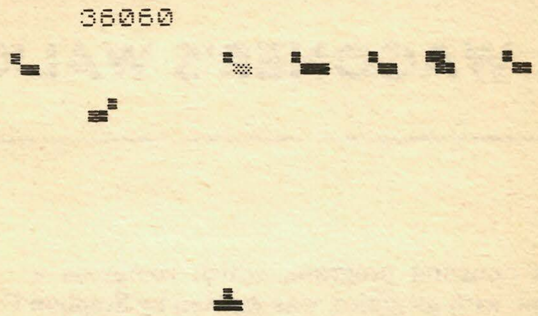
A number of strange little ducks fly overhead, some from right to left, and the others in the opposite direction. In this game by Peter Shaw your goal, needless to say, is to shoot down the ducks.

You move your shooting base from right to left using the **8** and **5** keys, to move in the direction shown by the arrows on those keys. You fire by pressing the **0** key. At the end of a round (when all the ducks have been shot) you'll be given a "marksman rating." There is a high-score feature, so you can try to better your rating from round to round. The rating is related to the number of shots it took you to kill all the ducks. There is a slight pause after one round, before a new one begins automatically.

```
  5 POKE 16389,0
  7 LET HS=0
 10 LET A=15
 15 LET SH=0
 20 LET SC=0
 30 LET A$=" ▀▒  ▀▄▄  ▀▄  ▀▚  ▀▄
▀▒ ▀▄ ▀▄
```

```
  40 LET B$="▄█    ▒▀ ▄▀    ▄▀    ▒▀
   ▄▀ ▄▀     "
  50 PRINT AT 2,0;A$,,B$
  60 IF A$="
              " AND B$="
                          " THEN GOT
O 3000
  80 LET A=A+(INKEY$="8")-(INKEY
$="5")
  90 PRINT AT 12,A-3;"  ▟▙  "
  95 IF INKEY$="0" THEN GOSUB 10
00
 100 LET A$=A$(3 TO )+A$(1 TO 2)
 110 LET B$=B$(32)+B$( TO 31)
 120 GOTO 50
 170 LET B$=B$(30 TO 32)+B$( TO
29)
1000 LET C=A-1
1005 LET SH=SH+7
1010 FOR B=10 TO 1 STEP -2
1020 LET A=A+(INKEY$="8")-(INKEY
$="5")
1030 PRINT AT 2,0;A$,,B$
1040 PRINT AT 12,A-3;"  ▟▙  ";AT
 B,C;"▌";AT B,C;" "
1060 LET A$=A$(6 TO )+A$(1 TO 5)
1075 PRINT AT 2,0;A$,,B$
1080 IF B=4 AND B$(C+2)<>" " THE
N GOSUB 1500
1090 IF B=2 AND A$(C+1)<>" " THE
N GOSUB 2000
1110 NEXT B
1120 RETURN
1500 LET B$(C+2)="█"
1510 LET SC=SC+2763
1520 PRINT AT 0,7;SC;AT 4,0;B$
1540 LET B$(C+1 TO C+3)="   "
1550 RETURN
2000 LET A$(C+1)="█"
2020 LET SC=SC+9741
2040 PRINT AT 2,0;A$;AT 0,7;SC
2050 LET A$(C TO C+2)="   "
2060 RETURN
3000 LET SC=INT (27394*SC/1+SH)
3010 IF SC>HS THEN LET HS=SC
3020 PRINT AT 0,0;"MARKSMAN RATI
NG IS ";SC
3030 PRINT AT 4,0;"BEST SO FAR I
S ";HS
3040 FOR G=1 TO 50
```

```
3050 NEXT G
3060 CLS
3070 GOTO 10
```



WAGONER'S WALK

This amusing program, which combines a race/bet theme with graphics, was written by Stephen Ormrod. You start the game with 20, and are attending a race meeting between four rather worn-out wagons. You see the wagons before the race, and can bet on one of them to win.

Lines 10 to 40 briefly explain the rules, while lines 50 to 250 initialize the graphics. The shapes are held in a string array, A. The wagons are displayed "in the paddock" by the routine from line 480 to line 570. Lines 480 to 815 deal with your bet. The maximum bet is either your credit level or $10, whichever is lower. The computer will not accept larger bets than this, nor will it accept bets lower than $1.00, or bets made on wagons which don't exist.

I'll leave you to see lines 820 to 905 in action, rather than explain them here. The four lanes are printed out by the routine from 1000, which also prints the start and finish lines, plus the wagons. The race itself is run by the lines 1158 to 1210. It will take a few minutes for a race to be run, and once it has, the screen will fill with the checkered flag, and a "bank statement" will

appear. You'll be given the chance of betting on another race, or of quitting with your winnings.

```
  10 PRINT "WAGONER""S WALK - RU
LES:"
  15 PRINT ,,"YOU HAVE $20. YOU
HAVE BEEN","INVITED TO A DAY AT
THE RACES","-SO WATCH YOUR MONEY
.THERE ARE"
  20 PRINT "4 WAGONS IN THE RACE
, ALL FROM ANEARBY SCRAPYARD - S
O THEY DO   TEND TO BE RATHER SL
OW. "
  25 PRINT ,,"YOU ARE INVITED TO
 BET ON","ONE OF THEM TO WIN. FI
RST,","HOWEVER, YOU MAY SEE THEM
 IN","THE PADDOCK."
  30 PRINT ,,"PRESS ""C"" TO CON
TINUE"
  35 IF INKEY$<>"C" THEN GOTO 35
  40 CLS
  45 LET CR=20
  50 DIM N$(4,4)
  60 LET N$(1)="JIM"
  70 LET N$(2)="JOE"
  80 LET N$(3)="JACK"
  90 LET N$(4)="JOHN"
 100 DIM A$(4,4,7)
 105 FOR N=1 TO 4
 110 LET A$(N,1)=" ███   "
 115 LET A$(N,3)=" ██████"
 120 LET A$(N,4)="▒▒O▒▒O▒"
 125 NEXT N
 130 LET A$(1,2)=" JIM █▄"
 135 LET A$(2,2)=" JOE █▄"
 140 LET A$(3,2)=" JACK█▄"
 145 LET A$(4,2)=" JOHN█▄"
 150 LET B$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒"
 155 LET C$="░░░░░░░░░░░░░░░░░░░░
░░░░░░░░░░"
 160 DIM A(4)
 170 LET A(1)=2
 180 LET A(2)=7
 190 LET A(3)=12
 200 LET A(4)=17
 210 DIM B(4)
```

```
 220 FOR N=1 TO 4
 230 LET B(N)=0
 240 NEXT N
 250 LET X$="▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
▚▚▚▚▚▚▚▚▚▚▚▚▚"
 480 PRINT AT 7,0;"WAGON:","DRIV
ER:"
 490 PRINT AT 3,0;C$;B$
 500 FOR N=1 TO 4
 502 PRINT AT 8+N,0;N,N$(N)
 505 FOR O=0 TO 31-(7*N)
 510 FOR P=1 TO 4
 515 PRINT AT 0+P-1,O;A$(N,P)
 520 NEXT P
 525 NEXT O
 535 NEXT N
 537 PRINT AT 15,0;
 540 PRINT ,,B$
 550 PRINT ,,"WHEN YOU HAVE SEEN
 ENOUGH OF","THEM, PRESS ""C"""
 560 IF INKEY$<>"C" THEN GOTO 56
0
 570 CLS
 580 PRINT "ALRIGHT, YOU HAVE $"
;CR
 590 PRINT " -REMEMBER, THE WINN
ER PAYS AT 2 TO 1 (+ A BONUS?);
BUT IF YOU","LOSE, YOUR STAKE IS
 DEDUCTED"
 600 PRINT ,,"PRESS THE NO. CORR
ESPONDING TO","THE WAGON YOU WIS
H TO BET ON,","AND PRESS ""ENTER
"""
 610 PRINT "(1,2,3 OR 4)"
 620 INPUT W
 630 IF W>0 AND W<5 AND W=INT W
THEN GOTO 700
 640 CLS
 650 PRINT "?-BUT WAGON ";W;" DO
ES NOT "," RUN IN THIS RACE. PLE
ASE","DO NOT TRY TO CHEAT THE","
BOOKIES."
 660 PRINT ,,
 670 GOTO 600
 700 CLS
 710 PRINT "YOU BACKED WAGON ";W
 720 PRINT "- ";N$(W);" WILL BE
PLEASED"
 730 PRINT ,,"BUT HOW MUCH DO YO
U WISH TO BET?(LIMIT:$";
```

```
 740 IF CR<10 THEN PRINT CR;
 750 IF CR>=10 THEN PRINT "10";
 760 PRINT ")"
 765 PRINT ,,"- SAME PROCEDURE A
S BEFORE"
 770 INPUT M
 775 CLS
 780 IF M<=0 THEN GOTO 800
 785 IF M>10 THEN GOTO 805
 790 IF M>CR THEN GOTO 810
 795 GOTO 820
 800 PRINT "-WHAT THE HECK ARE Y
OU PLAYING  AT ?"
 802 PRINT "- COMMON SENSE SHOUL
D TELL YOU  TO BET AT LEAST $1"
 803 PRINT "HOW MUCH DO YOU REAL
LY MEAN?"
 804 GOTO 770
 805 PRINT "DON""T BE GREEDY"
 806 GOTO 803
 810 PRINT "- BUT YOU HAVE ONLY
GOT $";CR
 815 GOTO 803
 820 PRINT "$";M;" BETTED"
 825 PRINT "THE RACE STARTS SHOR
TLY...","THE RACERS ARE TUNING U
P","THEIR ENGINES..."
 830 PRINT ,,B$,,
 835 FOR N=1 TO 100
 840 NEXT N
 845 FOR N=1 TO 4
 850 PRINT "PHUT..";
 855 FOR O=1 TO 50
 860 NEXT O
 865 NEXT N
 870 PRINT "BANGGGGG"
 875 FOR N=1 TO 50
 880 NEXT N
 885 CLS
 887 LET Y=INT (RND*5)+3
 888 LET Z=INT (RND*20)+60
 889 PRINT AT 10,0;"BONUS $";Y;"
 IF YOUR WAGON WINS","IN LESS TH
AN ";Z;" TIME UNITS"
 890 GOSUB 5000
 900 GOSUB 5000
 905 CLS
1000 FOR N=1 TO 21 STEP 5
1010 PRINT AT N,0;B$
1015 PRINT AT N-1,0;C$
```

```
1020 NEXT N
1030 FOR N=1 TO 21
1040 PRINT AT N,7;"S";AT N,30;"█
F"
1050 NEXT N
1060 GOSUB 1100
1070 GOTO 1150
1100 FOR N=1 TO 4
1110 FOR O=1 TO 4
1120 PRINT AT A(N)+(O-1),B(N);A$
(N,O)
1130 NEXT O
1135 IF B(N)=24 THEN GOTO 1300
1140 NEXT N
1145 RETURN
1150 LET T=0
1151 PRINT AT 0,0;"$";M;" ON ";W
;" ";N$(W)
1152 PRINT AT 4,10;"GET READY"
1153 GOSUB 5000
1154 PRINT AT 4,14;"SET  "
1155 GOSUB 5000
1156 PRINT AT 4,11;"O        "
1157 GOSUB 5000
1158 PRINT AT 4,10;"  "
1160 PRINT AT 0,15;"TIME ";T
1170 LET P=INT (RND*4)+1
1180 LET B(P)=B(P)+1
1190 GOSUB 1100
1200 LET T=T+1
1210 GOTO 1160
1300 LET X=N
1305 FOR N=1 TO 50
1310 PRINT AT 0,20;"A WINNER";AT
 0,20;"A WINNER"
1320 NEXT N
1330 POKE 16418,0
1340 FOR N=0 TO 22
1350 PRINT AT N,0;X$
1360 NEXT N
1370 PRINT "WINNER ";X;" ";N$(X)
;"   TIME ";T
1380 FOR N=1 TO 50
1390 NEXT N
1400 CLS
1410 POKE 16418,2
1420 IF X=W THEN GOTO 3000
1430 PRINT "*********** YOU LOST
 ***********"
```

```
1440 PRINT ,,"YOU MUST PAY YOUR
DEBT"
1450 PRINT AT 10,0;"YOU HAD:","$
";CR
1460 PRINT "YOU LOST:","$";M
1465 LET CR=CR-M
1470 PRINT ,,"YOU NOW HAVE:","$"
;CR
1480 IF CR<=0 THEN GOTO 7000
1490 IF CR>=100 THEN GOTO 6000
1500 PRINT ,,B$,,
1510 PRINT "PRESS ""C"" TO PLAY
AGAIN","...OR ""S"" TO QUIT"
1520 IF INKEY$="S" THEN GOTO 800
0
1530 IF INKEY$<>"C" THEN GOTO 15
20
1540 CLS
1550 LET P=INT (RND*5)+1
1560 GOTO 1560+(10*P)
1570 PRINT "GULLIBLE AREN""T YOU
?"
1575 GOTO 1650
1580 PRINT "FORTUNE FAVORS THE B
RAVE"
1585 GOTO 1650
1590 PRINT "O.K. -BUT YOU MUST R
ING HOME"," TO TELL YOUR HUSBAND
/WIFE","/MOTHER"
1595 GOTO 1650
1600 PRINT "TUT TUT - GAMBLING A
DDICT,  ARE YOU?"
1605 GOTO 1650
1610 PRINT "OH WELL, I""M GAME I
F YOU ARE"
1650 FOR N=1 TO 4
1651 LET B(N)=0
1652 NEXT N
1653 PRINT AT 15,0;"...BUT YOU D
ON""T GET TO SEE","THEM IN THE P
ADDOCK THIS TIME"
1659 GOSUB 5000
1660 GOSUB 5000
1665 GOSUB 5000
1670 GOTO 570
3000 PRINT "*********** YOU WON
***********"
3010 PRINT ,,"NOW YOU COLLECT YO
UR WINNINGS"
```

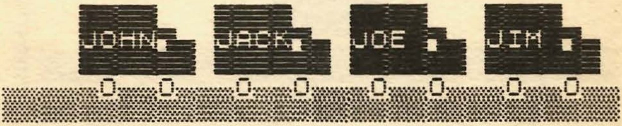
```
3020 PRINT AT 10,0;"YOU HAD:","$
";CR
3030 PRINT "WIN AT 2 TO 1:","$";
M*2
3040 PRINT "+ STAKE:","$";M
3050 LET CR=CR+(M*3)
3055 IF T<Z THEN GOTO 3100
3060 PRINT "NO TIME BONUS:","$0"
3070 GOTO 1470
3100 PRINT "TIME BONUS:","$";Y
3105 LET CR=CR+Y
3110 GOTO 1470
5000 FOR N=1 TO 25
5005 NEXT N
5010 RETURN
6000 PRINT ,,"YOU HAVE EXCEEDED
THE HOUSE","LIMITS OF $100 AND H
AVE BEEN"
6010 PRINT "FORCED TO RETIRE FRO
M THE GAME  WITH YOUR WEALTH","W
ILL YOU MARRY ME"
6020 STOP
7000 PRINT "YOU ARE SHORT. REMEM
BER YOU OWE THE COMPUTER $20. YO
U MAY LEAVE AN I.O.U."
7010 STOP
8000 CLS
8010 FOR N=1 TO 20
8020 PRINT "CHICKEN....",
8030 NEXT N
8040 PRINT AT 15,0;B$,,
8050 IF CR>20 THEN GOTO 8100
8060 IF CR=20 THEN GOTO 8200
8070 PRINT "- BUT YOU STILL OWE
THE COMPUTER£";20-CR
8080 STOP
8100 PRINT "- BUT YOU CAN FORFEI
T YOUR $";CR-20;"PROFIT AS A TIP
 TO THE COMPUTER"
8110 STOP
8200 PRINT "- BUT YOU ONLY BROKE
 EVEN"
```

WAGONEER"S WALK - RULES:

YOU HAVE $20. YOU HAVE BEEN
INVITED TO A DAY AT THE RACES
-SO WATCH YOUR MONEY.THERE ARE
4 WAGONS IN THE RACE, ALL FROM A
NEARBY SCRAPYARD - SO THEY DO
TEND TO BE RATHER SLOW.

YOU ARE INVITED TO BET ON
ONE OF THEM TO WIN. FIRST,
HOWEVER, YOU MAY SEE THEM IN
THE PADDOCK.

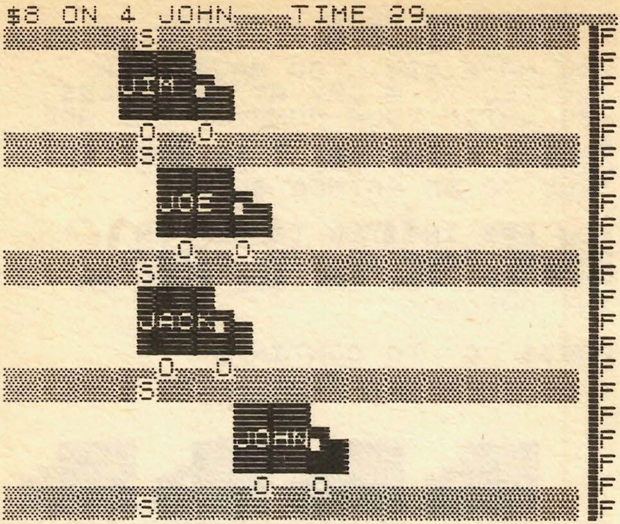
PRESS "C" TO CONTINUE



WAGON:	DRIVER:
1	JIM
2	JOE
3	JACK
4	JOHN

WHEN YOU HAVE SEEN ENOUGH OF
THEM, PRESS "C"

WAGONER"S WALK
S A ORMROD 7/1982



************ YOU WON ************

NOW YOU COLLECT YOUR WINNINGS

YOU HAD: $20
WIN AT 2 TO 1: $16
+ STAKE: $8
TIME BONUS: $4

YOU NOW HAVE: $48

PRESS "C" TO PLAY AGAIN
...OR "S" TO QUIT

Driving Games

GRAND PRIX

This game of skill, written by Jim Archer, combines steering, braking, and accelerating around a rather complex race course. The car is steered into a starting speed of 40 mph, from which you can accelerate up to a maximum of 200 mph, but it is quite difficult to complete the course at this speed without crashing at least once. Every crash costs you a time penalty of 10 seconds. There is a PAUSE statement within the main loop which is related to the current speed, so the program does actually get faster as your speed increases. At the final lap, the average time per lap is given in minutes and seconds, and you're graded as a driver from A to F. Only the best can get an A.

```
   5 REM "GRAND PRIX"
  10 PRINT " ** THE GRAND PRIX P
ROGRAM **"
  20 PRINT ,,,,"USE THE KEYS Q W
 E";TAB 13;"A   D";TAB 13;"Z X C
";TAB 8;"TO STEER YOURSELF,","KE
YS N,M TO BRAKE/ACCELERATE"
```

```
  25 PRINT ,,"WARNING: PRESSING
ANY OTHER KEY","  WILL STOP THE
CAR"
  30 PRINT ,,"HOW MANY LAPS? ";
  40 INPUT L1
  45 LET L=1
  50 PRINT L1
  54 DIM A$(20,32)
  55 DIM B$(20,50)
  57 FAST
  60 GOSUB 1000
  65 CLS
  68 LET YL=0
  70 FOR X=1 TO 20
  80 LET Z=1
  90 IF B$(X,Z)=" " THEN GOTO 14
0
 100 LET Y=VAL B$(X,Z TO Z+1)
 110 PRINT AT X,Y-1;"*"
 115 LET A$(X,Y)="*"
 120 LET Z=Z+2
 130 GOTO 90
 140 NEXT X
 160 LET S=0
 165 LET T=0
 170 LET U=100
 175 LET A$(14,28)="-"
 180 SLOW
 190 LET X=14
 200 LET Y=28
 210 PRINT AT X,Y-1;"█"
 220 PRINT AT 0,0;"READY..";
 230 FOR W=1 TO 3
 240 PRINT 4-W;"..";
 250 PAUSE 50
 260 NEXT W
 265 PRINT AT 0,0;"SPEED:0     TI
ME:0         LAP:1"
 270 LET F$="X"
 280 GOSUB 1300
 290 PRINT AT X,Y-1;A$(X,Y)
 300 LET X=X+I
 310 LET Y=Y+J
 320 PRINT AT X,Y-1;"█"
 322 IF A$(X,Y)<>" " THEN GOSUB
1500
 324 LET T=T+1+U/100
 325 PAUSE U
 327 LET S=2*(100-U)
```

```
 328 PRINT AT 0,6;S;"  ";AT 0,16
;T;"  "
 330 IF INKEY$="" THEN GOTO 290
 335 IF INKEY$="M" OR INKEY$="N"
 THEN GOTO 1400
 340 LET F$=INKEY$
 350 GOTO 280
 990 STOP
 999 REM COARSE DATA
1000 LET B$(1)="12131415"
1010 LET B$(2)="101116252627"
1020 LET B$(3)="0708091314172428
"
1030 LET B$(4)="0611121518222326
29"
1040 LET B$(5)="0405080910161920
21252730"
1050 LET B$(6)="0203071723242831
"
1060 LET B$(7)="0105061819202122
2931"
1070 LET B$(8)="0104121314151629
31"
1080 LET B$(9)="020411172932"
1090 LET B$(10)="02050809101113
1415183032"
1100 LET B$(11)="03050713161830
32"
1110 LET B$(12)="03050709101112
1516182021222324252627283032"
1120 LET B$(13)="03050710141819
293032"
1130 LET B$(14)="02050810141617
18212223242526272930 32"
1140 LET B$(15)="01040708101420
27293032"
1150 LET B$(16)="01030610151617
18192223242526272930 32"
1160 LET B$(17)="01040608092129
3032"
1170 LET B$(18)="02050608212324
25262728293032"
1180 LET B$(19)="03082132"
1190 LET B$(20)="04050607222324
25262728293031"
1195 RETURN
1300 LET I=(F$="Z")+(F$="X")+(F$
="C")-(F$="Q")-(F$="W")-(F$="E")
1310 LET J=(F$="E")+(F$="D")+(F$
```

```
="C")-(F$="Q")-(F$="A")-(F$="Z")
1320 IF F$<>"" AND U=100 THEN LE
T U=80
1330 IF I=0 AND J=0 THEN LET U=1
00
1340 RETURN
1400 IF INKEY$="M" AND U>=20 THE
N LET U=U-20
1410 IF INKEY$="N" AND U<=80 THE
N LET U=U+20
1430 GOTO 290
1500 IF A$(X,Y)="-" THEN GOTO 16
00
1505 LET U=100
1510 LET T=T+10
1515 IF S=0 THEN RETURN
1520 PRINT AT 21,5;"***CRASH***"
1530 PAUSE 60
1540 PRINT AT 21,5;"           "
1550 LET F$=""
1560 LET S=0
1570 LET I=0
1580 LET J=0
1590 RETURN
1600 IF S=0 THEN RETURN
1605 LET L=L+1
1607 IF L>L1 THEN GOTO 1630
1610 PRINT AT 0,27;L
1620 RETURN
1630 LET T=T/L1
1635 LET M=INT (T/60)
1640 LET S=INT (T-60*M+.5)
1650 PRINT AT 21,0;"AV/LAP:";M;"
:";
1660 IF S<10 THEN PRINT "0";
1670 PRINT S;" MIN:GRADE ";
1680 IF T<=105 THEN PRINT "A-CON
CEIT"
1690 IF T>105 AND T<=125 THEN PR
INT "B-FAST"
1700 IF T>125 AND T<=175 THEN PR
INT "C-AVERAGE"
1710 IF T>175 AND T<=200 THEN PR
INT "D-MEDIOCRE"
1720 IF T>200 AND T<=225 THEN PR
INT "E-SLOW"
1730 IF T>225 THEN PRINT "F-SNAI
L"
1750 STOP
```

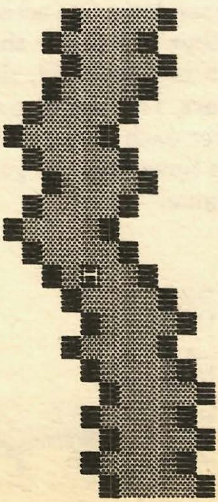
ALLEY DRIVER

In *Alley Driver*, written by Said Hassan, you have to drive a car down a constantly twisting track. Said explains: "The idea for the program is not really original, I know, but I think the way I've done it in this game is. Instead of scrolling the screen to give a racing-car effect, as is often used in these sorts of programs, the car (an inverse H) races down the screen. The effect, I feel, is slightly smoother and faster than scrolling.

"After each section is completed, the screen clears and a new track appears. The program supports a high-score feature, and after each game will ask the player if he or she wishes to have another game. Pressing Y will produce a new game."

```
  10 LET H=CODE ""
  20 LET S=CODE ""
  30 CLS
  40 LET X=CODE "▙"
  50 LET A=CODE "▙"
  60 FOR N=CODE "" TO CODE "="
  70 PRINT TAB A;"█▒▒▒▒▒▒▒█"
  80 LET A=A+(A<CODE "▛" AND A)*
SGN (RND-.5)+(NOT A)-(A=CODE "▛"
)
```

```
  90 NEXT N
 100 LET A=X
 110 FOR N=PI/PI TO CODE "="
 120 PRINT AT N,X;
 130 IF PEEK (PEEK 16398+256*PEE
K 16399)=CODE "▓" THEN GOTO 210
 140 PRINT AT N-PI/PI,A;"▒";AT N
,X;"■"
 150 LET A=X
 160 LET X=X+(INKEY$="0")-(INKEY
$="1")
 170 NEXT N
 180 CLS
 190 LET S=S+N
 200 GOTO CODE "2"
 210 CLS
 220 LET S=S+N
 230 PRINT ,,,"R.I.P.",,,"SCORE
=";S,
 240 IF H<S THEN LET H=S
 250 PRINT "HIGH SCORE=";H,,,"PL
AY AGAIN?"
 260 IF INKEY$="" THEN GOTO 260
 270 IF INKEY$="Y" THEN GOTO COD
E "="
```



SQUEEZER IV

Vroom, vroom... and you're away, driving your car along a constantly twisting and turning road. The road changes width as well as direction, demanding even more of your skill. The road will start off very wide (to get you used to the wheel) but will close in progressively. To move your car, use the 1 key (left) or the 0 key (right). The .731 in line 22 controls the rate at which the road narrows. Change the second or third decimal place to alter this. If you hit the side of the road, your score will appear, and you'll be offered another game. *Squeezer IV* was written by Nick Wilson. Lines 21 to 70 set up and then decrease the road width, and move the car. The routine from lines 80 to 150 asks if you want a new game, and from 202 the road is controlled.

```
10 REM SQUEEZER IV
11 REM NICK WILSON
12 LET B=0
13 LET X=16
14 CLS
19 LET M=0
20 LET L=0
```

```
  21 PRINT AT 21,M;"█";TAB (31-L
);"█"
  22 IF RND>.731 THEN LET L=L+1
  23 LET M=L
  25 IF L>13 THEN GOTO 200
  30 IF INKEY$="0" THEN LET X=X+
1
  40 IF INKEY$="1" THEN LET X=X-
1
  50 PRINT AT 17,X;
  55 IF PEEK (PEEK 16398+256*PEE
K 16399)<>0 THEN GOTO 80
  56 PRINT "▓"
  60 LET B=B+1
  65 SCROLL
  70 GOTO 21
  80 PRINT B
  90 GOTO 110
 100 CLS
 110 PRINT AT 10,10;"TRY AGAIN ?
"
 120 IF INKEY$="" THEN GOTO 120
 140 IF INKEY$="Y" THEN RUN
 150 STOP
 202 LET L=L+(1 AND RND>.5)-(1 A
ND RND>.5)
 203 IF L>26 OR L<0 THEN GOTO 20
2
 205 SCROLL
 210 PRINT TAB L;"█   █"
 230 IF INKEY$="0" THEN LET X=X+
1
 240 IF INKEY$="1" THEN LET X=X-
1
 250 PRINT AT 17,X;
 260 IF PEEK (PEEK 16398+256*PEE
K 16399)<>0 THEN GOTO 80
 270 LET B=B+1
 280 PRINT "▓"
 290 GOTO 202
```

Board Games and Simulations

SLOT MACHINE

Written by Adam Waring and Mike Cleverley, *Slot Machine* uses a flashy machine-code routine to reverse the display. It is called during the introduction, winning, and losing routines.

The object of the game is to win a grand total of $50. This is achieved by gambling on the one-armed bandit (see lines 40 to 80). It costs $1.00 per go, and you win $5 for getting two numbers the same, $15 for getting three the same.

The program has RESPIN and NUDGE routines. To save, type **GOTO 6550**, start your recorder, and then press **ENTER**. Upon loading, this program will start running on its own. Start by entering the following routine, which is used to put the machine code into the REM statement:

```
  1 REM 12345678901234567890
 10 INPUT X
 20 LET A$=""
 30 IF A$="" THEN INPUT A$
 40 IF A$="S" THEN STOP
 50 POKE X,16*CODE A$+CODE A$(2
TO )-476
 60 LET X=X+1
```

```
70 LET A$=A$(3 TO )
80 GOTO 30
```

After you've entered that, run it and input the following. The first prompt requires 16514 as entry, then:

```
2A, 0C, 40, 06,  17, 2B, 23, 7E,
FE, 76, 20, 03,  10, F8, C9, C6,
80, 77, 18, F2
```

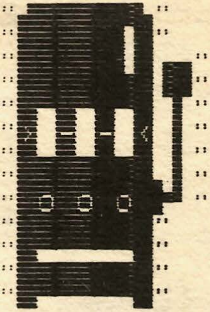
Line 1 should now look like this:

```
   1 REM E£RND▚*F7 SAVE TAN LEN
█?/ PAUSE
```

Once it does, enter as a direct command **POKE 16510,0** Then, enter the rest of the program:

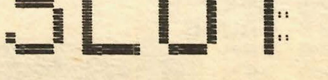
```
   0█REM E£RND▚*F7 SAVE TAN LEN
█?/ PAUSE
   2 REM
   3 REM      SLOT MACHINE
   5 REM   COPYRIGHT   (C)
   6 REM   MIKE CLEVERLEY
   7 REM          AND
   8 REM      ADAM WARING
   9 REM
  20 GOSUB 2000
  24 LET P=PEEK 16396+PEEK 16397
*256
```

```
  25 RAND
  27 DIM A(4)
  30 LET A=5
  34 REM
  35 REM  PRINT MACHINE
  36 REM
  38 CLS
  40 PRINT
  43 PRINT
  46 PRINT
  50 PRINT
  53 PRINT
  56 PRINT
  60 PRINT
  63 PRINT
  70 PRINT
  73 PRINT
  76 PRINT
  78 PRINT
  79 PRINT
```



```
  80 REM
  89 REM
  90 REM  DISPLAY CASH
  91 REM
 100 GOSUB 3000
 120 PRINT AT 15,9;"YOU HAVE $";
A;
 125 IF A<>INT A THEN PRINT 0
 130 PRINT TAB 5;"PRESS ""C"" TO
 CONTINUE",TAB 9;"""Q"" TO QUIT"
 140 IF INKEY$="Q" THEN STOP
 150 IF INKEY$<>"C" THEN GOTO 14
0
 180 GOSUB 3000
 181 REM
 182 REM  NUMBERS FOR REELS
 183 REM
 185 FOR L=1 TO 3
 190 FOR M=1 TO 10-L*2
 195 FOR N=L TO 3
 200 LET A(N)=INT (RND*10)
 203 POKE P+166+N*2-1,A(N)+28
 205 NEXT N
 206 NEXT M
 207 PAUSE L
 208 REM
 209 REM  RESPIN OR NUDGE?
 210 REM
 211 LET G=RND
```

```
 212 IF G<.7 THEN GOTO 4000
 215 IF G<.8 THEN GOTO 400
 240 REM
 250 REM  RESPIN
 260 REM
 270 LET X=INT (RND*3)*2+1
 303 PRINT AT 16,0;"DO YOU WANT
A RESPIN?"
 307 PRINT AT 8,X;"█"
 311 PRINT AT 8,X;"▒"
 313 IF INKEY$="N" THEN GOTO 400
0
 315 IF INKEY$<>"Y" THEN GOTO 30
7
 320 FOR N=1 TO 20
 324 LET A((X+1)/2)=INT (RND*10)
 326 POKE P+166+X,A((X+1)/2)+28
 328 NEXT N
 350 GOTO 4000
 400 REM
 401 REM  NUDGE
 402 REM
 420 LET G=INT (RND*4+2)
 430 PRINT AT 15,3;"YOU HAVE ";G
;" NUDGES"
 450 FOR N=1 TO G
 460 IF INKEY$="0" THEN GOTO 400
0
 470 IF INKEY$<"1" OR INKEY$>"3"
 THEN GOTO 460
 480 LET B=CODE INKEY$-28
 490 LET A(B)=A(B)-1
 495 IF A(B)<0 THEN LET A(B)=9
 500 PRINT AT 5,1;A(1);TAB 3;A(2
);TAB 5;A(3)
 510 NEXT N
 520 GOTO 4000
2010 REM
2011 REM  INTRODUCTION
2012 REM
2050 PRINT
2100 PRINT "
2110 PRINT "
2120 PRINT "
2130 PRINT "
```



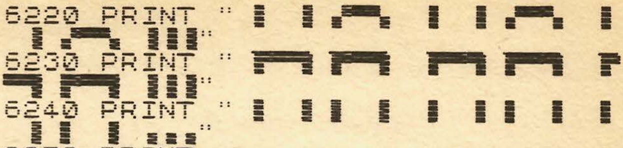
```
2140 PRINT
2150 PRINT "[graphics]"
2160 PRINT "[graphics]"
2170 PRINT "[graphics]"
2180 PRINT "[graphics]"
2185 LET B$="MIKE CLEVERLEY AND
ADAM WARING"
2190 LET A$="COMPLETED ON 19TH J
UNE 1982 BY"
2205 FOR N=1 TO 30
2210 PRINT AT 11,N;A$(N)
2215 PRINT AT 12,31-N;B$(31-N)
2220 FOR M=1 TO 5
2230 NEXT M
2240 NEXT N
2273 FOR N=1 TO 21
2275 RAND USR 16514
2276 FOR M=1 TO 7
2277 NEXT M
2278 NEXT N
2280 LET A$="
INSTRUCTIONS?"
2294 PRINT AT 14,0;A$
2300 LET A$=A$(2 TO )+A$(1)
2305 IF INKEY$="N" THEN RETURN
2310 IF INKEY$<>"Y" THEN GOTO 22
90
2311 REM
2312 REM INSTRUCTIONS
2313 REM
2320 CLS
2330 PRINT "INSTRUCTIONS"
2340 PRINT "------------"
2360 PRINT "TRY YOUR HAND AT THE
 ONE ARMED  BANDIT. WIN A TOTAL
OF $50."
2370 PRINT "YOU ARE GIVEN $5 TO
START OFF"
2380 PRINT "WITH. EACH SPIN COST
S $1. YOU   SPIN BY PRESSING ""C
""."
2390 PRINT "GETTING 2 REELS THE
SAME WINS   YOU $5. GETTING 3 TH
E SAME WINS"
```

```
2400 PRINT "$15. DURING THE GAME
 YOU MAY GETA RESPIN. THESE ARE
FREE, AND"
2410 PRINT "THE FLASHING BUTTON
INDICATES   WHICH REEL MAY BE RE
SPUN. YOU"
2420 PRINT "RESPIN BY PRESSING T
HE ""Y"" KEY, IF YOU DO NOT WISH
 TO RESPIN"
2430 PRINT "THEN PRESS THE ""N""
 KEY."
2431 PRINT
2432 PRINT
2433 PRINT
2435 PRINT "PRESS ""C"" T CONTIN
UE"
2436 IF INKEY$<>"C" THEN GOTO 24
36
2437 CLS
2440 PRINT "NUDGES ARE ALSO AVAI
LABLE AT    RANDOM STAGES THROUG
HOUT THE"
2450 PRINT "GAME. YOU WILL HAVE
FROM 2 TO 5 NUDGES AT A TIME. YO
U PRESS THE"
2460 PRINT "COLUMN NUMBER ""1"",
 ""2"" OR ""3"", TO NUDGE THE A
PPROPRIATE COLUMN."
2470 PRINT "TO STOP NUDGING, PRE
SS ""0"" ."
2480 PRINT "THE GAME MAY BE TERM
INATED AT   ANY TIME BY PRESSING
 ""Q"" OR "" "" ."
2483 PRINT
2484 PRINT
2485 PRINT
2490 PRINT "PRESS ""C"" TO CONTI
NUE"
2495 IF INKEY$<>"C" THEN GOTO 24
95
2500 RETURN
3000 REM
3002 REM  CLEAR SCREEN
3006 REM
3020 FOR N=13 TO 21
3030 PRINT AT N,0;"
                  "

3060 NEXT N
3070 RETURN
```

```
4000 REM ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
4002 REM ▒ MONEY WON AND LOST ▒
4004 REM ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
4010 GOSUB 3000
4050 LET A=A-1
4060 LET A(4)=A(1)
4100 FOR N=1 TO 3
4110 IF A(N)=A(N+1) THEN GOSUB 4
500
4120 NEXT N
4400 IF A>50 THEN GOTO 5000
4410 IF A<1 THEN GOTO 6000
4490 GOTO 80
4500 FOR M=13 TO 21
4520 PRINT AT M,1;"00000"
4530 PRINT AT M-1,1;"     "
4570 NEXT M
4580 LET A=A+5
4590 RETURN
5000 REM ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
5002 REM ▒ WOW YOUVE WON ▒
5004 REM ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
5009 CLS
5050 PRINT "▄█▄  ▄█▄  ▄█▄  ▄█▄  ▄█▄"
5060 PRINT "█▙▄  █▙▄  █▙▄  █▙▄  █▙▄"
5070 PRINT "▗██  ▗██  ▗██  ▗██  ▗██"
5080 PRINT "▀█▀  ▀█▀  ▀█▀  ▀█▀  ▀█▀"
5090 PRINT
5100 PRINT "▚ ▞ ▛▀▀▌ ▌  ▌▞▌ ▌▛▀▀"
5110 PRINT " ▚▞ ▌  ▌▌  ▌ ▌ ▚ ▞▛▀▀"
5120 PRINT "  ▌ ▚▄▄▞ ▚▄▄▞  ▚ ▞▙▄▄"
5125 PRINT
5130 PRINT "▛▀▀▌▛▀▀▌▛▀▀▌▌ ▞▛▀▀"
5140 PRINT "▛▀▀▌▛▀▛▌▌ ▌▛▀▌▛▀▀"
5150 PRINT "▙▄▄▞▌ ▚ ▚▄▄▞▌ ▚▙▄▄"
5160 PRINT
```

```
5170 PRINT "                     :: ::
Q"" TO QUIT"
5180 PRINT "                     :: ::
S"" TO START"
5190 PRINT "                   "
5200 PRINT
5210 PRINT "
::
5220 PRINT "                        "
5230 PRINT "
::
5300 RAND USR 16514
5304 FOR N=1 TO 10
5306 NEXT N
5310 IF INKEY$="S" THEN RUN 21
5320 IF INKEY$<>"Q" THEN GOTO 53
00
5330 PRINT TAB 8;"GOODBYE SUCKER
"
5500 STOP
6000 REM
6010 REM  YOURE A BORN LOSER
6020 REM
6030 CLS
6050 PRINT
6060 PRINT
6100 PRINT "
           ::
6110 PRINT "
           ::
6120 PRINT "
       ::
6130 PRINT "
           ::
6140 PRINT
6150 PRINT
6160 PRINT "
           ::
6170 PRINT "
           ::
6180 PRINT "
       ::
6190 PRINT "
           ::
6200 PRINT
6210 PRINT
```



```
6250 PRINT
6260 PRINT
6270 PRINT "  ""S"" TO START",""
"Q"" TO QUIT"
6300 GOTO 5300
6550 SAVE "BANDIT"
6560 RUN
```

REACTION

This program is based on an old reaction test for pilots and bombadiers. In its original form, when computers were steam-powered, the test consisted of flashing lights and switches, controlled by tubes. The T/S 1000 has done away with all that.

You'll see an interesting display on the screen at the start of the program, made up of the numbers 1 to 9 down the left-hand side of the screen. A black bar finds its way down the screen, until it reaches the bottom, followed by a series of other bars. The aim of the test is to stop the bar as close to the top of the screen as possible, by pressing the number or letter that the bar is on. This may sound simple, but you'll soon discover that it is not. The score, shown at the top of the screen, fluctuates according to how fast you can get to the bar, so it's quite possible for the score to fall below zero. You can speed the whole thing up by adjusting the value of N in line 110. When the score gets above 49, the test is terminated. *Reaction* comes from Nick Wilson.

```
 10 REM REACTION TEST
 11 REM  NICK WILSON
 12 FAST
 13 PRINT
 20 FOR I=1 TO 20
 30 PRINT CHR$ (I+28);" ";
 32 FOR J=1 TO 10
 33 PRINT "---";
 34 NEXT J
 40 PRINT
 50 NEXT I
 55 SLOW
 56 LET S=10
 60 LET J=INT (RND*26)+5
 61 PRINT AT 0,0;"SCORE = ";S;"
"
 70 LET Y=1
 75 FOR I=1 TO 20
 76 LET Y=I
 80 PRINT AT Y+1,J;"██";AT Y,J;
"--"
 85 LET N=0
 86 IF INKEY$<>"" THEN GOTO 150
 90 LET A$=INKEY$
100 LET N=N+1
110 IF N=4 THEN GOTO 150
120 IF A$="" THEN GOTO 90
130 IF A$=CHR$ (I+29) THEN GOTO
200
150 NEXT I
160 PRINT AT Y+1,J;"  ";
165 LET S=S-20
170 GOTO 60
200 LET S=S+(20-I)
205 IF S>=50 THEN STOP
206 PRINT AT Y+1,J;"---"
210 GOTO 60
```

CHECKERS SEVEN

This game is somewhat like checkers, except that it is played on a 7 × 7 board. The pieces move like checkers pieces—diagonally one square, jumping over an opponent for a capture into an empty square beyond the opponent. The main differences from checkers, apart from the size of the board, are that pieces are able to move forward and backward at will—every piece can move like a king—and there are no multiple jumps. The T/S 1000 is the X's moving down the screen, and you are the O's. You move by entering the number of the square from which you are moving—entering the number along the left-hand edge first, then the number across the top, then pressing **ENTER**. The T/S 1000 will keep track of the score, tell you before it moves the move it intends to make, and terminate the game as soon as one player manages to capture five of the opponent's pieces.

```
 5 FAST
30 GOSUB 9000
40 GOSUB 7000
50 SLOW
60 GOSUB 8000
```

```
  70 GOSUB 6000
  90 GOSUB 8000
 100 FAST
 110 PRINT AT 0,0;
1000 GOTO 40
6000 PRINT AT 2,0;"FROM?"
6060 INPUT M
6070 PRINT AT 2,0;M;" TO?"
6080 INPUT N
6085 PRINT AT 2,0;"          "
6087 PRINT AT 0,0;
6090 LET H(N)=52
6095 IF ABS (M-N)=22 OR ABS (M-N
)=18 THEN LET H((M+N)/2)=26
6096 IF ABS (M-N)=22 OR ABS (M-N
)=18 THEN LET ME=ME+1
6100 LET H(M)=26
6990 RETURN
7010 FOR A=76 TO 12 STEP -1
7020 IF H(A)<>61 THEN GOTO 7060
7030 FOR B=1 TO 4
7032 IF A<28 AND B<3 THEN GOTO 7
050
7033 IF A>60 AND B>2 THEN GOTO 7
050
7035 LET Q=2*Z(B)
7040 IF H(A+Z(B))=52 AND H(A+Q)=
26 THEN GOTO 7070
7050 NEXT B
7060 NEXT A
7065 GOTO 7300
7070 LET H(A+Z(B))=26
7080 LET H(A)=26
7090 LET H(A+Q)=61
7092 LET Y=A+Q
7093 LET X=A
7095 LET IT=IT+1
7100 RETURN
7310 LET Y=0
7320 LET Y=Y+1
7330 LET K=INT (RND*66)+12
7340 IF H(K)<>61 AND Y<100 THEN
GOTO 7320
7350 IF H(K)<>61 THEN GOTO 7460
7360 FOR T=1 TO 4
7370 IF H(K+Z(T))=26 THEN GOTO 7
400
7380 NEXT T
7390 IF Y<70 THEN GOTO 7310
7395 GOTO 7460
```

```
7400 LET H(K+Z(T))=61
7410 LET H(K)=26
7415 LET X=K
7417 LET Y=K+Z(T)
7420 RETURN
7460 PRINT "I CONCEDE"
7470 STOP
8000 PRINT
8010 PRINT "I MOVED FROM ";X;" T
O ";Y
8015 PRINT
8020 PRINT "SCORES: YOU: ";ME;"
ME: ";IT
8040 PRINT
8050 PRINT TAB 8;" 1234567"
8055 PRINT TAB 8;"█████████"
8060 FOR J=70 TO 10 STEP -10
8061 LET A=H(J+1)
8062 LET B=H(J+2)
8063 LET C=H(J+3)
8064 LET D=H(J+4)
8065 LET E=H(J+5)
8066 LET F=H(J+6)
8077 LET G=H(J+7)
8080 PRINT TAB 7;J/10;"█";CHR$ (
A);CHR$ (B);CHR$ (C);CHR$ (D);CH
R$ (E);CHR$ (F);CHR$ (G);"█";J/1
0
8090 NEXT J
8100 PRINT TAB 8;"█████████"
8110 PRINT TAB 8;" 1234567"
8120 IF IT=5 OR ME=5 THEN GOTO 8
140
8130 RETURN
8140 IF IT=5 THEN PRINT "I WIN"
8150 IF ME=5 THEN PRINT "YOU WIN
"
8990 STOP
9000 LET IT=0
9010 LET ME=0
9020 DIM H(99)
9025 DIM Z(4)
9030 FOR A=1 TO 99
9050 IF A>77 OR A=70 OR A=60 OR
A=68 OR A=69 OR A=50 OR A=59 OR
A=58 OR A=40 OR A=49 OR A=48 THE
N GOTO 9090
9055 IF A=30 OR A=38 OR A=39 OR
A=20 OR A=28 OR A=29 OR A<11 THE
N GOTO 9090
```

```
9060 LET H(A)=26
9070 IF A=72 OR A=74 OR A=76 OR
A=61 OR A=63 OR A=65 OR A=67 THE
N LET H(A)=61
9080 IF A=21 OR A=23 OR A=25 OR
A=27 OR A=12 OR A=14 OR A=16 THE
N LET H(A)=52
9090 NEXT A
9100 LET Z(1)=-11
9110 LET Z(2)=-9
9120 LET Z(3)=11
9130 LET Z(4)=9
9140 RETURN
```

```
I MOVED FROM 45 TO 34

SCORES: YOU: 2 ME: 1

     1234567
    
  7 ;X;;;;; 7
  6 ;;;;X;; 6
  5 ;X;X;X; 5
  4 ;;O;;;; 4
  3 ;O;;;O; 3
  2 O;;;;;O 2
  1 ;O;;;;; 1
    
     1234567
```

BETWEEN THE STARS

When the delights of Earth begin to pall, you may hunger to soar into darkest space. You have been given responsibility for the security of a cube of space, measuring 10 × 10 × 10. The Terran Federation, sparing no expense in the defense of Earth, has provided you with a spaceship equipped with a T/S 1000 as its onboard computer. Now it is your turn to guard the space lanes.

There are a lot of things demanding your attention. Your position within the cube is given by the three co-ordinates under the line **SHIP IS CRUISING AT THE CO-ORDINATES:** The first co-ordinate is your position north/south (with lower numbers to the south), the second is your position across the cube—i.e., east/west—and the third is your position within the cube (forward/back). You can see that the ability to visualize in three dimensions is useful.

The alien craft is moving very slowly within the cube, but although you know, at all times, its direction from you, you do not know how far away it is. You have to hit it as many times as you can before the time counter decrements to zero, and without colliding with the alien craft. Running out of energy will also ter-

minate the game. You will know you are close enough to fire when the computer reports that the alien ship is firing at you. Every hit decrements your energy supply rather drastically.

The game is simple to play, despite the bewildering amount of input the program is giving you. You just touch the key which refers to the direction you want to move: **N**, **S**, **E**, or **W** to move north, south, east, or west, **A** to advance, **R** to retreat; and **L** to fire your laser at the alien ship. If, for example, you know the ship is to the north, you can just hold down the **N** key until you move onto the same north/south plane as the ship, then test for proximity by firing.

You'll find that the program will teach you how to play the game. Just keep in mind that you have to get as close as possible to the alien ship to fire, and that your task is to get as many on your "tally" as possible before the game ends.

Between the Stars was written by Roger MacIntyre.

```
  10 REM BETWEEN THE STARS
  20 REM BY ROGER MACINTYRE
  30 GOSUB 1070
  40 GOSUB 800
  50 IF L<0 THEN GOTO 500
  80 PRINT AT 17,0;"ENTER YOUR C
OMMAND"
  90 PRINT AT 18,2;"N,E,S,W,(L)A
SER,",",(A)DVANCE,(R)ETREAT"
 100 LET L=L-0.25
 120 IF INKEY$="" THEN GOTO 120
 130 IF INKEY$="L" THEN GOSUB 32
0
 140 IF INKEY$="N" THEN LET X=X-
1
 150 IF INKEY$="S" THEN LET X=X+
1
 160 IF INKEY$="E" THEN LET Y=Y+
1
 170 IF INKEY$="W" THEN LET Y=Y-
1
```

```
 180 IF INKEY$="A" THEN LET Z=Z-
1
 190 IF INKEY$="R" THEN LET Z=Z+
1
 195 PRINT AT 5,0;S$
 200 GOSUB 620
 210 IF RND>0.5 THEN GOTO 40
 240 LET A=A+INT ((RND*3)-(RND*3
))
 250 IF A<1 THEN LET A=1
 255 IF A>10 THEN LET A=10
 260 LET B=B=INT ((RND*3)-(RND*3
))
 265 IF B>10 THEN LET B=10
 266 IF B<1 THEN LET B=1
 270 IF RND>0.5 THEN GOTO 40
 280 LET C=INT ((RND*3)-(RND*3))
 290 IF C<1 THEN LET C=1
 300 IF C>10 THEN LET C=10
 310 GOTO 40
 320 REM ** FIRE LASER **
 330 LET L=L-0.75
 340 PRINT AT 1,0;
 480 LET T=T+1
 482 FOR J=1 TO 50
 483 NEXT J
 485 PRINT AT 1,0;T$
 490 RETURN
 500 PRINT
 510 PRINT TAB 3;"█ TERMINATION
█"
 520 PRINT
 530 IF TI<0 THEN PRINT "WE HAVE
 BEEN IN SPACE TOO LONG"
 540 IF L>0 THEN PRINT "█ WE HAV
E BEEN DEFEATED █"
 550 PRINT
 555 PRINT AT 10,0;"ENERGY LEFT
▒ ";L;" ERGS    "
 560 IF L<=0 THEN PRINT "███████
 ENERGY BANKS EMPTY █████"
 570 STOP
 580 PRINT
 590 PRINT "WE HAVE COLLIDED WIT
H THE";TAB 8;"ALIEN SHIP"
 610 STOP
 620 REM ** ALIENS SHOOT **
 630 IF ABS (A-X)>3 OR ABS (B-Y)
>3 OR ABS (C-Z)>3 THEN RETURN
 650 IF RND>0.75 THEN RETURN
```

```
 660 PRINT AT 1,0;"▒▒▒▒▒█ ALIENS
FIRING AT US █▒▒▒▒▒"
 670 FOR J=1 TO 50
 680 NEXT J
 690 PRINT AT 1,0;T$
 700 IF RND>0.7 THEN GOTO 770
 710 PRINT AT 1,0;"█ ALIEN FIRE
HAS HIT US █"
 720 LET L=L-7
 730 IF L<=0 THEN GOTO 500
 740 FOR J=1 TO 50
 750 NEXT J
 760 RETURN
 770 PRINT AT 1,0;"▌▌▌▌▌▌▌▌ALIEN
FIRE MISSED▐▐▐▐▐▐▐▐"
 780 FOR J=1 TO 50
 790 NEXT J
 792 PRINT AT 1,0;T$
 795 RETURN
 800 REM ** PRINT OUT **
 850 PRINT AT 10,0;"ENERGY LEFT
█ ";L;" ERGS    "
 870 LET TI=TI-1
 880 IF TI=0 THEN GOTO 500
 890 PRINT AT 19,20;"TIME: ";TI
 900 IF L<3 THEN PRINT AT 12,4;"
▗▀█ ENERGY LOW █▀▖"
 920 PRINT AT 20,19;"TALLY: ";T
 930 PRINT AT 14,0;"█ SHIP IS CR
UISING AT THE"
 935 PRINT "COORDINATES:"
 936 PRINT TAB 4;X;"  ";Y;"  ";Z
;"  "
 940 IF A=X AND B=Y AND C=Z THEN
 GOTO 580
 960 PRINT AT 5,0;"▗▀▖";AT 5,0;"▀▖
▗▀";AT 5,0;"   "
 967 PRINT AT 5,0;"ALIEN CRAFT I
S MOVING"
 970 IF A<>X OR B<>Y THEN PRINT
"TO THE ";
 980 IF A<X THEN PRINT "NORTH";
 990 IF A>X THEN PRINT "SOUTH";
1000 IF B>Y THEN PRINT "EAST";
1010 IF B>Y THEN PRINT "WEST";
1020 IF C=Z THEN PRINT " OF US"
1030 IF C>Z THEN PRINT " BEHIND
US"
1040 IF C<Z THEN PRINT " IN FRON
T OF US"
```

```
1060 RETURN
1070 REM ** INITIALIZE **
1090 LET L=INT (RND*10)+1
1100 LET T=0
1110 LET TI=35
1140 LET A=R(RND*10)+1
1150 LET B=R(RND*10)+1
1160 LET C=R(RND*10)+1
1170 LET X=R(RND*10)+1
1180 LET Y=R(RND*10)+1
1190 LET Z=R(RND*10)+1
1195 LET S$="

            "
1197 LET T$="
            "
1200 FOR J=0 TO 63
1205 PLOT J,0
1210 PLOT J,43
1220 NEXT J
1230 FOR J=0 TO 4
1240 PLOT 0,J
1250 PLOT 63,J
1260 NEXT J
1270 RETURN
```

DRAGON'S GOLD

The aim of *Dragon's Gold* is simple: to accumulate as much gold as possible, while wandering through a complex maze of tunnels, caves, and doors, and to avoid the dragon and mineshafts. You enter **A** to move ahead, **L** to move left, or **R** to move right. Entering a space before pressing **ENTER** will cause the game to stop. *Dragon's Gold* was written by D. C. Owen.

```
  1 REM DRAGONS GOLD
  2 REM BY D C OWEN 1982
  3 RAND
  5 LET G=0
 50 SCROLL
 51 SCROLL
 52 SCROLL
 53 SCROLL
 55 PRINT TAB 8;"DRAGON/S GOLD"
 56 SCROLL
 57 SCROLL
 58 SCROLL
 60 PRINT "YOU HAVE:-"
 65 SCROLL
 70 PRINT G;" BLOCKS OF GOLD"
 80 SCROLL
 90 SCROLL
100 PRINT "AHEAD OF YOU IS A";
```

```
 120 GOSUB 1000
 125 LET B$=A$
 127 SCROLL
 130 PRINT "ON THE LEFT IS A";
 140 GOSUB 1000
 145 SCROLL
 150 LET L$=A$
 155 PRINT "AND ON THE RIGHT IS
A";
 160 GOSUB 1000
 165 LET R$=A$
 170 SCROLL
 171 SCROLL
 172 PRINT "WHICH WAY DO YOU WAN
T TO GO?"
 173 SCROLL
 174 SCROLL
 175 PRINT "A - AHEAD"
 176 SCROLL
 177 PRINT "L - LEFT"
 178 SCROLL
 179 PRINT "R - RIGHT"
 180 INPUT K$
 185 SCROLL
 186 SCROLL
 190 IF K$="A" AND B$="D" THEN G
OSUB 2000
 200 IF K$="R" AND R$="D" THEN G
OSUB 2000
 210 IF K$="A" AND B$="T" THEN G
OSUB 3000
 220 IF K$="L" AND L$="T" THEN G
OSUB 3000
 230 IF K$="R" AND R$="T" THEN G
OSUB 3000
 240 IF K$="A" AND B$="C" THEN G
OSUB 4000
 250 IF K$="L" AND L$="C" THEN G
OSUB 4000
 260 IF K$="R" AND R$="C" THEN G
OSUB 4000
 270 IF NOT (K$="L" OR K$="R" OR
 K$="A") THEN GOTO 170
 280 GOTO 50
 990 REM **************
1000 GOTO 1000+INT (RND*3+1)*100
1100 PRINT " DOOR"
1110 LET A$="D"
1120 RETURN
1200 PRINT " TUNNEL"
```

```
1210 LET A$="T"
1220 RETURN
1300 PRINT " CAVE"
1310 LET A$="C"
1320 RETURN
1999 REM ************
2000 REM ** DOOR **
2010 GOTO 2000+INT (RND*4+1)*100
2100 PRINT "IT IS LOCKED. MOVE O
N"
2110 RETURN
2200 LET Q=INT (RND*9+1)*100
2210 PRINT "IT WILL OPEN. THERE"
2215 SCROLL
2220 PRINT "ARE ";Q;" GOLD BLOCK
S IN HERE"
2230 LET G=G+Q
2240 RETURN
2300 PRINT "THERE IS A LAKE HERE
. YOU"
2305 SCROLL
2310 PRINT "CANNOT SEE THE FAR S
IDE."
2315 SCROLL
2320 PRINT "ARE YOU GOING TO TRY
"
2330 SCROLL
2340 PRINT "AND CROSS IT?"
2345 SCROLL
2350 INPUT C$
2360 SCROLL
2370 IF CODE (C$)<>CODE "Y" THEN
 RETURN
2380 LET K=INT (RND*3)+1
2381 SCROLL
2382 IF K=2 THEN PRINT "YOU HAVE
 ESCAPED WITH"
2383 SCROLL
2384 IF K=2 THEN PRINT G;" BLOCK
S OF GOLD"
2386 IF K<>2 THEN PRINT "UNFORTU
NATELY, YOU HAVE"
2388 SCROLL
2390 IF K<>2 THEN PRINT TAB 10;"
DROWNED..."
2395 STOP
2400 LET K=INT (RND*9+1)*50
2405 SCROLL
2410 PRINT "THIS ROOM CONTAINS A
 DRAGON"
```

```
2415 SCROLL
2420 PRINT "IT DEMANDS ";K;" GOL
D BLOCKS"
2425 SCROLL
2430 PRINT "OR IT WILL EAT YOU"
2440 FOR J=1 TO 20
2450 SCROLL
2460 PRINT TAB J;"STAND BY"
2470 NEXT J
2475 SCROLL
2480 IF G>K THEN PRINT "YOU HAVE
 ENOUGH"
2485 SCROLL
2490 IF G<K THEN PRINT "...BUT Y
OU HAVEN/T GOT"
2495 SCROLL
2500 IF G<K THEN PRINT "ENOUGH..
.SO BYE BYE";END
2510 LET G=G-K
2520 RETURN
2999 REM ************
3000 REM ** TUNNEL **
3010 IF RND>0.85 THEN RETURN
3015 SCROLL
3020 PRINT "YOU HAVE ESCAPED"
3025 SCROLL
3030 PRINT "WTH ";G;" GOLD BLOCK
S"
3040 STOP
3999 REM ************
4000 REM ** CAVE **
4005 SCROLL
4010 GOTO 4000+INT (RND*3+1)*100
4100 PRINT "THE CAVE IS EMPTY."
4105 SCROLL
4110 PRINT TAB 8;"MOVE ON"
4120 RETURN
4200 LET Q=INT (RND*10+1)*100
4210 PRINT "THERE ARE ";Q;" GOLD
 BLOCKS"
4215 SCROLL
4220 PRINT "HERE TO ADD TO YOUR
STORE"
4230 LET G=G+Q
4240 RETURN
4300 IF RND>0.9 THEN GOTO 4400
4301 FOR H=1 TO 24
4305 SCROLL
4307 NEXT H
4310 PRINT "OH NO"
```

```
4315 SCROLL
4320 FOR J=1 TO 15
4330 PRINT TAB 2*J;"██"
4335 SCROLL
4340 NEXT J
4345 SCROLL
4350 PRINT "IT IS A MINESHAFT...
"
4355 SCROLL
4360 PRINT "YOU ARE DEAD"
4370 STOP
4400 SCROLL
4405 PRINT "THERE ARE NOISES AHE
AD"
4407 SCROLL
4410 PRINT "DO YOU WANT TO INVES
TIGATE?"
4420 INPUT K$
4430 IF CODE K$<>CODE "Y" THEN R
ETURN
4440 GOTO 4000
```

HIT IT

Seven boxes appear at the top of the screen, numbered (rather logically, one would think) from 1 to 7. The message **GET READY** flashes for a couple of seconds, and then vanishes. Following a random delay, a black square appears in one of the boxes. You have just over a second to hit the corresponding number's key. (To make it easier, change the 20 in line 130 to a bigger number.)

If you got the number right, the message **HIT IT...** will appear. If you failed, you'll either get **TOO LATE** if you took too long, or **WRONG** if you were wrong. Keep watching, because another number will soon appear. We suggest you plan to take the best of, say, five games, and keep a tally of which player gets the most points. The graphics in the program are:

Line 17 : E7R

Line 26 : 5 B

Line 40: W 6 Q

Line 100: space

Lines 12 to 45 print and number the seven boxes, while lines 46 to 70 flash the **GET READY** message. The box is chosen and printed by the routine from 90,

and lines 140 to 165 choose the correct message to display. *Hit It* was written by Nick Wilson.

```
  10 REM   HIT  IT
  11 REM NICK WILSON
  12 PRINT "   1   2   3   4   5
   6   7"
  13 PRINT
  15 PRINT "  "
  16 FOR I=1 TO 7
  17 PRINT "▛▀▜ ";
  18 NEXT I
  20 PRINT "  "
  25 FOR I=1 TO 7
  26 PRINT "▌ ▐ ";
  27 NEXT I
  28 PRINT "  "
  30 FOR I=1 TO 7
  40 PRINT "▙▄▟ ";
  45 NEXT I
  46 FOR I=1 TO 3
  50 PRINT AT 10,10;"GET READY"
  55 LET L=SIN 5
  60 PRINT AT 10,10;"GET READY"
  65 LET L=SIN 5
  70 NEXT I
  75 LET C=0
  80 PRINT AT 10,10;"          "
  90 LET W=INT (RND*7)+1
  91 FOR I=1 TO (RND*30)
  92 NEXT I
 100 PRINT AT 3,(W*3)+W-1;"█"
 110 LET A$=INKEY$
 120 LET C=C+1
 130 IF C=20 THEN GOTO 210
 140 IF A$="" THEN GOTO 110
 150 IF CODE A$<CODE "1" OR CODE
 A$>CODE "7" THEN GOTO 230
 160 IF VAL A$=W THEN PRINT AT 1
0,10;"HIT IT..."
 165 IF VAL A$<>W THEN GOTO 230
 170 FOR I=1 TO 30
 180 NEXT I
 190 PRINT AT 0,0;
 200 RUN
 210 PRINT AT 10,10;"TOO LATE."
```

```
215 PRINT AT 0,0;
220 RUN
230 PRINT AT 10,10;"WRONG...."
235 PRINT AT 0,0;
240 RUN
```

TIC TAC TOE

I guess there is no need to tell you how to play this game. You and the computer take it in turns, trying to get three O's or three X's in a row. Unlike many computer versions of the game, this program allows you to win now and then. Most of the computer *Tic Tac Toe* games are unbeatable, with a draw being the best you can do.

You move by entering the number of the square into which you want to move. You enter your move first, and the computer will ignore you if you do not enter 5 as your first move, the center square.

This version of the game is capable of handling over 40,000 developments of *Tic Tac Toe*, about an eighth of the possible games which can occur.

```
 1 REM TIC TAC TOE
 2 REM ADAPTED T HARTNELL
 3 REM FROM ZX80 PROGRAM
 4 REM BY STUART ROBERTS
10 DIM B(9)
20 DIM P(9)
30 FOR A=1 TO 9
40 LET B(A)=A
50 NEXT A
```

```
  70 LET E=0
  80 LET Q=0
  90 LET N=0
 100 LET X=4
 110 PRINT AT 5,0;
 120 FOR A=1 TO 9
 130 IF A=X THEN GOSUB 770
 140 IF B(A)=0 THEN GOTO 400
 150 IF B(A)=10 THEN GOTO 430
 160 PRINT B(A);" ";
 170 NEXT A
 180 PRINT
 190 PRINT
 200 IF N=1 THEN GOTO 460
 210 IF E=8 THEN PRINT " I WIN "
 215 IF E=8 THEN GOTO 970
 220 IF Q=5 THEN PRINT " ITS A D
RAW "
 225 IF Q=5 THEN GOTO 970
 230 INPUT Z
 240 IF B(Z)<>Z THEN GOTO 230
 250 LET Q=Q+1
 260 IF Z=11 THEN GOTO 300
 270 LET N=1
 280 LET B(Z)=0
 290 GOTO 100
 300 LET B(5)=10
 310 GOTO 90
 320 FOR A=C TO D
 330 IF B(A)=A THEN LET B(A)=10
 340 LET A=A+F
 350 NEXT A
 360 GOTO 90
 370 IF B(5)=5 THEN GOTO 300
 380 LET B(D)=10
 390 GOTO 90
 400 PRINT "O ";
 410 LET P(A)=1
 420 GOTO 170
 430 PRINT "X ";
 440 LET P(A)=4
 450 GOTO 170
 460 LET H=810
 470 LET G=0
 480 LET C=1
 490 LET D=9
 500 LET F=3
 510 GOSUB H
 520 LET C=3
```

```
530 LET D=7
540 LET F=1
550 GOSUB H
560 LET D=9
570 LET F=2
580 GOSUB H
590 LET C=2
600 LET D=8
610 GOSUB H
620 LET C=1
630 LET D=7
640 GOSUB H
650 LET D=3
660 LET F=0
670 GOSUB H
680 LET C=4
690 LET D=6
700 GOSUB H
710 LET C=7
720 LET D=9
730 GOSUB H
740 LET G=G+1
750 IF G=5 THEN GOTO 90
760 GOTO 480
770 LET X=X+3
780 PRINT
790 PRINT
800 RETURN
810 LET E=0
820 FOR A=C TO D
830 LET E=E+P(A)
840 LET A=A+F
850 NEXT A
860 IF E=3 THEN GOTO 960
870 IF G=0 THEN RETURN
880 IF E=8 THEN GOTO 320
890 IF G=1 THEN RETURN
900 IF E=2 THEN GOTO 320
910 IF G=2 THEN RETURN
920 IF E=5 THEN GOTO 320
930 IF G=3 THEN RETURN
940 IF E=1 OR E=4 THEN GOTO 370
950 RETURN
960 PRINT " YOU WIN "
970 PRINT
980 PRINT
990 PRINT "DO YOU WANT ANOTHER GAME?",,"(Y OR N)"
1000 INPUT K$
```

```
1005 CLS
1010 IF K$<>"N" THEN RUN
1200 PRINT
1210 PRINT "OK, THANKS FOR PLAYI
NG"
```

```
 1 2 3

 4 5 6

 7 8 9
```

```
 1 2 3

 4 O 6

 7 8 9
```

```
 1 2 3

 4 O 6

 7 8 X
```

```
 O 2 3

 4 O 6

 X 8 X
```

```
O X 3
4 O 6
X O X
```

```
O X 3
O O X
X O X
```

```
1 2 3
4 5 6
7 8 9
```

```
1 2 3
4 5 6
7 8 9
```

MUSIC(?)

This program is called *Music* (?) because certain musicians may claim the definition of music is not wide enough to stretch to the output of the computer in this program.

The sound, which can be quite musical, is produced through the television speaker. You may have to tune the TV slightly off the optimum position for the picture to hear the sound at its best.

This program, by Tim Hartnell, produces music at random, with a particular note being produced by each of the Z loops. For additional notes, of different pitches, all you need to do is add extra subroutines at the end, and modify line 18 to allow for them.

Notice how line 18 takes the place of the ON...GOTO command available in many dialects of BASIC. Line 18 takes the place of all of the following lines:

```
IF K = 5 THEN GOTO 220
IF K = 4 THEN GOTO 180
IF K = 3 THEN GOTO 140
IF K = 2 THEN GOTO 100
IF K = 1 THEN GOTO 60
IF K = 0 THEN GOTO 20
```

You should keep this programming technique in mind when you are getting short of memory. Tim Hartnell's program is followed by a shortened version of it by Ken Mahogany.

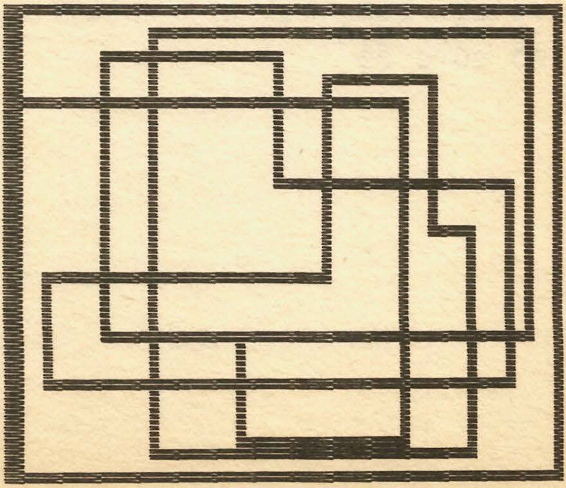
```
  10 REM        *MUSIC(?)*
  15 REM (C) HARTNELL 1982
  16 LET K=INT (RND*6)
  17 REM NOTE HOW NEXT LINE
       WORKS AS AN "ON...GOTO"
  18 GOTO (220 AND K=5)+(180 AND
 K=4)+(20 AND K=0)+(60 AND K=1)+
(100 AND K=2)+(140 AND K=3)
  20 FOR Z=1 TO 10*(RND*3)
  40 FAST
  50 NEXT Z
  55 IF RND>.5 THEN RUN
  70 SLOW
  95 IF RND>.5 THEN RUN
 100 FOR Z=1 TO RND*60
 110 SLOW
 120 FAST
 130 NEXT Z
 135 IF RND>.5 THEN RUN
 140 FOR Z=1 TO 10*(RND*80)
 150 SLOW
 160 FAST
 170 NEXT Z
 175 IF RND>.5 THEN RUN
 180 FOR Z=1 TO 10*(RND*80)
 190 SLOW
 200 FAST
 210 NEXT Z
 215 IF RND>.5 THEN RUN
 220 FOR Z=1 TO 10*(RND*80)
 230 SLOW
 240 FAST
 250 NEXT Z
 260 RUN
```

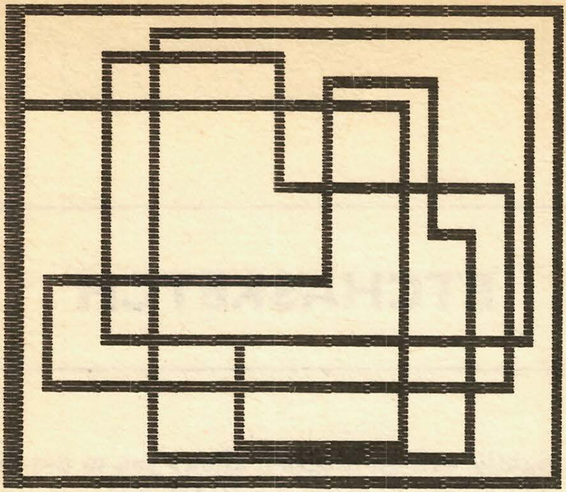
```
 10 REM *MUSIC
         (C)
   MAHOGANY 1982
 20 FOR Z=1 TO RND*60
 30 SLOW
 40 FAST
 50 NEXT Z
 60 FOR Z=1 TO RND*60
 70 SLOW
 80 FAST
 90 NEXT Z
100 FOR Z=1 TO RND*60
110 SLOW
120 FAST
130 NEXT Z
140 FOR Z=1 TO RND*60
150 SLOW
160 FAST
170 NEXT Z
180 PAUSE RND*10
190 RUN
```

ETCHASKETCH

This tightly written program allows you to use the **5**, **6**, **7**, and **8** keys to move the **PLOT** blob around the screen, drawing pictures of your choice.

Once you've got it running, try and modify it to give you a choice of starting position, and/or to "turn off" the blob from time to time to move it to a new position on the screen without leaving a trail.





```
 10 LET A=VAL "1"
 20 LET B=A
 30 LET A$=INKEY$
 40 IF A$="" THEN GOTO 30
 60 LET A=A+(A$="7")-(A$="6")
 70 LET B=B+(A$="8")-(A$="5")
 80 PLOT B,A
100 GOTO 30
```

LIFE

The game of *Life* was invented by John Conway of Cambridge University, England, in October 1970. It simulates the birth, death, and growth of cells in a closed colony.

Before the state of a cell for the "next generation" (a generation is a complete check and reprint of the grid upon which the colony lives) is determined, it must be compared with the eight surrounding cells. If there are two or three occupied cells around the one being checked, and the one being checked is occupied, there is no change; it survives till the next generation. If there are three and only three occupied cells, and the cell being checked is empty, a cell is "born" there in the next generation. If there are four or more neighbors, the cell being checked "dies"—that is, is emptied in the next generation.

That is almost all the information you need to construct a game of *Life* from first principles. There is just one more thing—the rules are applied all over the grid at once, so you need one array to hold the current generation, and another to hold the new generation, so that changes for the next generation do not affect cells which have not yet been checked in the present generation.

Set up a 10 × 10 grid, and try and work out a program to (a) place some cells on it; (b) check each of these cells in turn in accord with Conway's laws, and then update a reference array; (c) copy the reference array into a "printout" array; and (d) print out the colony and start again.

Here's one way of doing it, which uses two "data" statements in the form of strings which are accessed element by element. A$ in line 30 contains information regarding the numerical relationship of cells to each other (e.g., +1 is one to the right, −1 is one to the left, and so on). A$ in line 90 is the position of the starting cells, when the grid is numbered 1 to 100. Line 30 contains the following: minus sign, plus sign, equals sign, pound sign, graphic from the **S** key, graphic from the **2** key, graphic from the **1** key, space.

Note that there is a comma after the last element within **A$** in line 90. This is needed for the "data" routine to work.

Other starting colonies you can try:

Beehive: 45, 45, 46, 64, 65, 66, 74, 76, 85

Cross: 43, 47, 54, 56, 65, 74, 76, 83, 87

Möbius: 23,24,25,33,34,35,43,44,45,56,57,58,66,67,68,76,77,78

Russian: 33,34,35,36,37,38,47,56,65,74,83,84,85,86. 87,88

Flame: 16,26,36,46,51,52,53,54,55,56,57,58,59,66,76,86,96

```
10 FAST
20 DIM E(8)
30 LET A$="-+=£▒█  "
40 FOR A=1 TO 8
50 LET E(A)=CODE A$(A)-11
60 NEXT A
70 DIM A(120)
80 DIM L(120)
```

```
 90 LET A$="64,55,65,75,76,46,5
6,76,86,"
 100 FOR A=1 TO LEN A$ STEP 3
 110 LET A(VAL A$(A TO A+1))=1
 120 LET L(VAL A$(A TO A+1))=1
 130 NEXT A
 140 LET GENERATION=0
 145 SLOW
 150 GOTO 310
 160 LET GENERATION=GENERATION+1
 170 FOR U=0 TO 9
 180 FOR B=1 TO 9
 190 LET F=U+10*B+2
 200 LET H=0
 210 FOR T=1 TO 8
 220 LET H=H+A(F+E(T))
 230 NEXT T
 240 IF A(F)=1 AND H<>3 AND H<>2
 THEN LET L(F)=0
 250 IF A(F)=0 AND H=3 THEN LET
L(F)=1
 260 NEXT B
 270 NEXT U
 275 SLOW
 280 FOR M=1 TO 100
 290 LET A(M)=L(M)
 300 NEXT M
 310 PRINT AT 5,0;
 320 FOR U=1 TO 9
 330 PRINT TAB 3;
 340 FOR B=0 TO 9
 350 LET F=U+10*B+1
 360 PRINT CHR$ A(F);" ";
 370 NEXT B
 380 NEXT U
 390 PRINT AT 3,10;"GENERATION "
;GENERATION
 400 FOR G=1 TO 100
 410 NEXT G
 420 FAST
 430 GOTO 160
```

```
GENERATION 0

 ≡       ≡
   ≡   ≡
     ≡
   ≡   ≡
 ≡       ≡
```

```
GENERATION 1

   ≡ ≡ ≡
   ≡   ≡
   ≡ ≡ ≡
```

```
GENERATION 2

     ≡
   ≡   ≡
 ≡       ≡
   ≡   ≡
     ≡
```

```
GENERATION 3

     ≡
   ≡ ≡ ≡
 ≡ ≡   ≡ ≡
   ≡ ≡ ≡
     ≡
```

```
GENERATION 4

  ■ ■ ■
■       ■
■       ■
■       ■
  ■ ■ ■
```

```
GENERATION 0

    ■
  ■ ■ ■
■ ■   ■ ■
```

```
GENERATION 1

  ■ ■ ■
■       ■
■ ■   ■ ■
```

```
GENERATION 2

    ■
  ■ ■ ■
■       ■
■ ■   ■ ■
```

```
GENERATION 3

    ≡ ≡ ≡
    ≡ ≡ ≡
  ≡       ≡
  ≡ ≡   ≡ ≡
```

```
GENERATION 4
      ≡
    ≡   ≡
  ≡       ≡
  ≡       ≡
  ≡ ≡   ≡ ≡
```

```
GENERATION 5
      ≡
    ≡ ≡ ≡
  ≡ ≡   ≡ ≡
≡ ≡       ≡ ≡
  ≡ ≡   ≡ ≡
```

```
GENERATION 6

    ≡ ≡ ≡
  ≡       ≡
≡           ≡
≡           ≡
≡ ≡ ≡   ≡ ≡ ≡
```

```
GENERATION 0

  ≡ ≡ ≡
≡   ≡   ≡
  ≡ ≡ ≡
```

```
GENERATION 1

    ≡
  ≡ ≡ ≡
≡       ≡
  ≡ ≡ ≡
    ≡
```

```
GENERATION 2

  ≡ ≡ ≡
  ≡ ≡ ≡
≡       ≡
  ≡ ≡ ≡
  ≡ ≡ ≡
```

```
GENERATION 3

    ≡
  ≡   ≡
≡       ≡
≡       ≡
≡       ≡
  ≡   ≡
    ≡
```

TENBY

Tenby is a relatively simple gambling game played with two dice, based on craps. To play, you roll two dice and add up their pips. If you roll a seven or an eleven on the first roll it is called a "natural," and you win, ending that round. Rolling a two, three, or twelve on the first roll is a disaster—the round ends immediately. Rolling four, five, six, eight, nine, or ten on the first roll becomes your "point." The aim of the game—assuming it has not ended with the first roll—is to roll your point again before you throw a seven.

The program keeps a tally of your wins and losses. If you like, you can modify the game to allow for betting, either with one player, the player and the computer, or two players. The percentage "ahead" you are is shown. If this is a negative number you are—needless to say—behind, rather than ahead.

```
10 REM TENBY
20 LET G=0
30 LET L=0
40 LET W=0
50 LET G=G+1
```

```
  60 PRINT AT 9,4;"GAME NUMBER "
;G
  70 PRINT AT 4,4;"WINS: ";W;" L
OSSES: ";L
  75 IF L>0 THEN PRINT AT 20,0;"
YOU ARE AHEAD BY ";INT ((W-L)*10
00/L)/10;" PERCENT"
  80 GOSUB 200
  90 IF A=7 OR A=11 THEN GOTO 33
0
 100 IF A=2 OR A=3 OR A=12 THEN
GOTO 340
 110 LET P=A
 120 PRINT AT 14,4;"YOUR POINT I
S ";P
 130 GOSUB 200
 140 IF A=P THEN GOTO 330
 150 IF A=7 THEN GOTO 340
 160 FOR T=1 TO 50
 170 NEXT T
 180 GOTO 130
 200 PRINT AT 0,0;"PRESS D TO RO
LL DICE";AT 0,6;"D"
 210 IF INKEY$<>"D" THEN GOTO 20
0
 220 PRINT AT 0,0;"
                "
 230 LET A=INT (RND*6+1)+INT (RN
D*6)+1
 240 FOR T=1 TO 50
 250 NEXT T
 260 PRINT AT 0,0;"YOU ROLLED ";
A
 270 RETURN
 330 FOR T=1 TO 20
 332 PRINT AT 7,8;"YOU WIN"
 333 PRINT AT 7,8;"YOU WIN"
 334 NEXT T
 335 LET W=W+1
 336 GOTO 370
 340 FOR T=1 TO 20
 342 PRINT AT 7,8;"YOU LOSE"
 344 PRINT AT 7,8;"YOU LOSE"
 345 NEXT T
 350 LET L=L+1
 370 FOR T=1 TO 100
 380 NEXT T
 390 CLS
 400 GOTO 50
```

BATTLE

This program by Chris Callender places you on a checkerboard measuring 9 × 8, in which you move diagonally, and capture by landing on top of an opponent. There are no multiple jumps.

The computer will have the first move in each game, and the aim of the *Battle* is to capture six of your opponent's pieces before he, she, or it manages to do so with your pieces.

You're playing from the bottom of the screen (O) and the computer from the top (X). You move by entering the number down the side relating to the square you're moving from, and the square across the top or bottom, as a single two-digit number, then—after pressing **NEWLINE**—the two-digit number representing the square you're moving to. Illegal moves will be rejected.

```
   5 REM BATTLE
   6 DIM B$(10,11)
  10 LET B$(1)="▒123456789▒"
  20 LET B$(2)="1X█X█X█X█X1"
  30 LET B$(3)="2█X█X█X█X█2"
  40 LET B$(4)="3X█X█ █X█X3"
  50 LET B$(5)="4█ █ █X█ █4"
  60 LET B$(6)="5 █ █ █ █ 5"
  70 LET B$(7)="6█O█O█O█O█6"
  80 LET B$(8)="7O█O█O█O█O7"
  90 LET B$(9)="8█O█O█O█O█8"
 100 LET B$(10)="▒123456789▒"
 102 LET HS=0
 105 LET CS=0
 110 DIM S$(12,13)
 115 FOR A=1 TO 10
 117 LET C$=B$(A)
 120 FOR B=1 TO 11
 125 LET S$(A,B)=C$(B TO )(1)
 130 NEXT B
 132 NEXT A
 135 IF RND>.5 THEN GOTO 150
 140 LET S$(5,5)="X"
 141 LET S$(5,7)=" "
 150 GOSUB 1130
 155 PRINT
 157 IF CS=6 THEN PRINT "I WIN "
;SW
 160 PRINT AT 15,0;"FROM?"
 165 INPUT MOVE
 166 PRINT AT 15,4;" ";MOVE;" TO
?"
 167 LET A=INT (MOVE/10)
 168 LET B=MOVE-10*A
 170 INPUT MOVE
 171 PRINT AT 15,0;"
        "
 172 LET C=INT (MOVE/10)
 175 LET D=MOVE-10*C
 180 IF ABS (A-C)<>1 OR ABS (B-D
)<>1 THEN GOTO 160
 190 IF S$(C+1)(D+1)="X" THEN LE
T HS=HS+1
 210 LET S$(A+1)(B+1)=" "
 220 LET S$(C+1)(D+1)="O"
 240 GOSUB 1130
 245 IF HS=6 THEN PRINT "YOU WIN
";SW
 250 LET A$="O"
 255 GOSUB 1000
```

```
 257 IF FL=1 THEN GOTO 300
 260 LET A$=" "
 265 GOSUB 1000
 300 LET S$(E)(F)=" "
 305 IF S$(E+G)(F+H)="O" THEN LE
T CS=CS+1
 310 LET S$(E+G)(F+H)="X"
 320 GOTO 150
1000 LET E=2
1001 LET F=2
1002 LET G=0
1003 LET H=0
1010 LET FL=0
1020 IF S$(E)(F)<>"X" THEN GOTO
1100
1040 IF S$(E+1)(F+1)=A$ OR S$(E+
1)(F-1)=A$ THEN LET G=1
1050 IF S$(E+1)(F+1)=A$ OR S$(E-
1)(F+1)=A$ THEN LET H=1
1060 IF S$(E-1)(F+1)=A$ OR S$(E-
1)(F-1)=A$ THEN LET G=-1
1070 IF S$(E+1)(F-1)=A$ OR S$(E-
1)(F-1)=A$ THEN LET H=-1
1080 IF G<>0 AND H<>0 THEN LET F
L=1
1085 IF FL=1 THEN RETURN
1100 LET E=E+1
1101 IF E>10 THEN LET F=F+1
1102 IF E>10 THEN LET E=2
1110 IF F>11 THEN RETURN
1120 GOTO 1010
1129 STOP
1130 PRINT AT 0,0;
1132 FOR A=1 TO 10
1135 PRINT
1140 FOR B=1 TO 11
1145 PRINT S$(A,B);
1150 NEXT B
1155 NEXT A
1160 PRINT
1165 PRINT
1170 PRINT "ME: ";CS;"   YOU: ";
HS
1180 RETURN
```

```
▒123456789▒
1X█X█X█X█X1
2█X█X█X█X█2
3 █X█ █X█X3
4█ █ █ █ █4
5 █X█ █O█ 5
6█ █ █ █O█6
7O█O█O█O█O7
8█O█O█O█O█8
▒123456789▒

ME: 2    YOU: 1

FROM 75 TO?
```

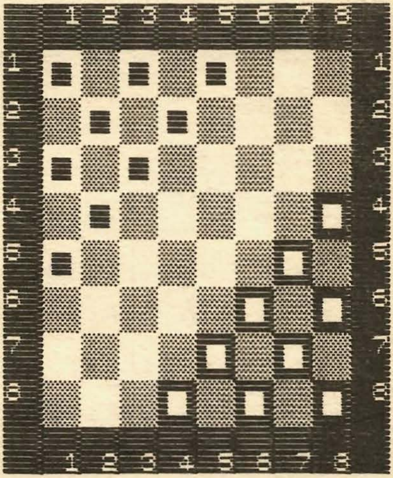
MANDALA CHECKERS/ CHOPPER CHECKERS

This program follows the standard rules of checkers, except that you play by starting in the corners of the board, rather than at the ends, and there are no multiple jumps and no kings. Any piece may move in any diagonal direction. Captures are as in checkers, by jumping over an opponent's piece into an empty square, always moving on the diagonal. Once you've entered *Mandala Checkers* in line with the first listing, you can easily modify it to play *Chopper Checkers*, which is closer to ordinary checkers. In *Chopper*, you move from left to right across the board, while the computer moves from right to left. You play again as in checkers, except you can move in any diagonal direction as if you had a board of kings, there are no kings, and there are no multiple jumps. You move in both games by entering your move as a single four-digit number (like 3344), which means you're moving from square 33 (the number across the top, then the number down the side) to square 44.

You'll find that this program, which uses four character cells for each square on the board, produces a

most effective display, which almost fills the screen. The board is printed at the start of the game, and from then on only the squares which change are reprinted, so it plays very rapidly.

The following are the only line numbers that differ between the two programs: 6176, 6440, 7090, 8070, 8080, 9010, 9020, 9040, 9105, 9190.



First we have *Mandala Checkers*:

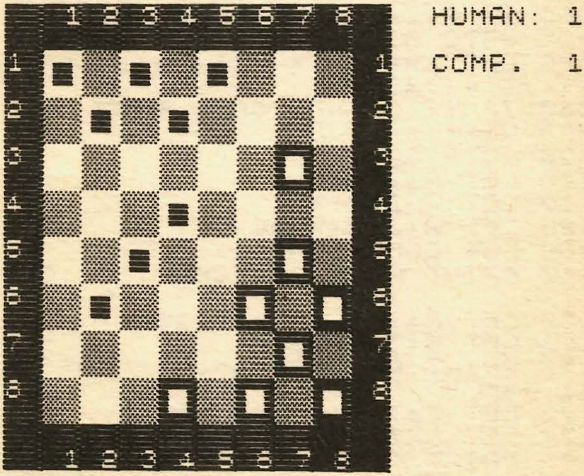
```
   20 GOSUB 9000
   30 GOSUB 8000
   35 SLOW
   40 GOSUB 7000
   50 GOSUB 6000
   60 IF HUM=7 THEN PRINT AT 19,0
;"YOU WIN ";W
   70 IF COMP=7 THEN PRINT AT 19,
0;"I WIN ";W
```

```
  80 GOTO 40
6000 REM **COMPUTER MOVE**
6010 FOR Z=88 TO 11 STEP -1
6020 IF A(Z)=C THEN GOTO 6050
6030 NEXT Z
6040 GOTO 6200
6050 LET Y=-11
6055 IF Z+Y>88 OR Z+Y<11 OR Z+2*Y>88 OR Z+2*Y<11 THEN GOTO 6070
6060 IF A(Z+Y)=H AND A(Z+2*Y)=E THEN GOTO 6100
6070 LET Y=-9*(Y=-11)+9*(Y=-9)+10*(Y=9)+(Y=100)
6080 IF Y<>0 THEN GOTO 6055
6085 NEXT Z
6090 GOTO 6200
6100 LET A(Z)=E
6110 LET A(Z+Y)=E
6120 LET A(Z+2*Y)=C
6130 LET COMP=COMP+1
6140 PRINT AT 2,22;"COMP.  ";COMP
6150 LET F=INT ((Z+Y)/10)
6160 LET G=Z+Y-10*F
6170 PRINT AT 2*G,2*F;"  ";AT 2*G+1,2*F;"  "
6172 LET F=INT ((Z+2*Y)/10)
6174 LET G=Z+2*Y-10*F
6176 PRINT AT 2*G,2*F;"▛▜";AT 2*G+1,2*F;"▙▟"
6180 LET F=INT (Z/10)
6182 LET G=Z-10*F
6184 PRINT AT 2*G,2*F;"  ";AT 2*G+1,2*F;"  "
6190 RETURN
6200 REM **NON-CAPTURE MOVE**
6210 FOR Z=1 TO 200
6220 LET K=INT (RND*78)+11
6230 IF A(K)=C THEN GOTO 6260
6240 NEXT Z
6250 GOTO 6500
6260 LET Y=-11
6280 IF A(K+Y)=E THEN GOTO 6330
6290 LET Y=-9*(Y=11)+9*(Y=-9)+11*(Y=9)+(Y=100)
6300 IF Y<>0 THEN GOTO 6270
6310 NEXT Z
6320 GOTO 6500
6330 IF K+2*Y>88 OR K+2*Y<11 THE
```

```
N GOTO 6400
6340 IF A(K+2*Y)=H THEN GOTO 624
0
6350 IF K-2*Y<11 OR K-2*Y>88 THE
N GOTO 6400
6360 IF A(K-2*Y)=H THEN GOTO 624
0
6400 LET A(K+Y)=C
6410 LET A(K)=E
6420 LET F=INT ((K+Y)/10)
6430 LET G=K+Y-10*F
6440 PRINT AT 2*G,2*F;"▛▜";AT 2*
G+1,2*F;"▙▟"
6450 LET F=INT (K/10)
6460 LET G=K-10*F
6470 PRINT AT 2*G,2*F;"  ";AT 2*
G+1,2*F;"  "
6480 RETURN
6500 FOR G=1 TO 200
6510 LET K=INT (RND*78)+11
6520 IF A(K)=C THEN GOTO 6600
6530 NEXT G
6540 PRINT AT 0,0;"I CONCEDE THE
 GAME"
6550 STOP
6600 IF A(K-11)=E THEN LET Y=-11
6610 IF A(K-11)=E THEN GOTO 6400
6620 IF A(K-9)=E THEN LET Y=-9
6630 IF A(K-9)=E THEN GOTO 6400
6640 GOTO 6540
7000 REM **PLAYER MOVE**
7010 PRINT AT 20,0;"ENTER YOUR M
OVE AS ""3344"""
7020 INPUT A$
7030 IF LEN A$<>4 THEN GOTO 7020
7040 PRINT AT 20,0;"
                    "
7050 LET A1=VAL A$(1)
7055 LET A2=VAL A$(2)
7060 LET B1=VAL A$(3)
7065 LET B2=VAL A$(4)
7070 LET A(10*B1+B2)=H
7080 LET A(10*A1+A2)=E
7090 PRINT AT 2*B2,2*B1;"▗▖";AT
2*B2+1,2*B1;"▝▘"
7100 PRINT AT 2*A2,2*A1;"  ";AT
2*A2+1,2*A1;"  "
7110 IF ABS (A1-B1)=1 THEN RETUR
N
```

```
7120 LET HUM=HUM+1
7125 LET A((10*B1+B2+10*A1+A2)/2
)=E
7130 PRINT AT (A2+B2),(A1+B1);"
 ";AT A2+B2+1,A1+B1;"  "
7140 PRINT AT 0,22;"HUMAN: ";HUM
7990 RETURN
8000 REM PRINT BOARD - START
8040 FOR Z=8 TO 1 STEP -1
8060 FOR X=1 TO 8
8070 IF A(10*Z+X)=H THEN PRINT A
T 2*X,2*Z;" ▄ ";AT 2*X+1,2*Z;"▀ "
8080 IF A(10*Z+X)=C THEN PRINT A
T 2*X,2*Z;"▛▜";AT 2*X+1,2*Z;"▙▟"
8090 IF A(10*Z+X)=B THEN PRINT A
T 2*X,2*Z;"▒▒";AT 2*X+1,2*Z;"▒▒"
8110 NEXT X
8120 NEXT Z
8130 RETURN
8990 STOP
8999 STOP
9000 FAST
9005 DIM A(100)
9010 LET H$="111315222431334251"
9020 LET C$="888677688475665748"
9030 LET B$="1214161821232527323
43638414345475254565861636567727
4767881838587"
9040 LET E$="8273645546372817263
544536271"
9050 FOR Z=1 TO 100
9060 LET A(Z)=9
9070 NEXT Z
9080 LET H=CODE "H"
9090 LET C=CODE "C"
9095 LET B=CODE "█"
9100 LET E=CODE " "
9105 FOR Z=1 TO 9
9110 LET A(VAL H$( TO 2))=H
9120 LET A(VAL C$( TO 2))=C
9130 LET H$=H$(3 TO )
9135 LET C$=C$(3 TO )
9140 NEXT Z
9150 FOR Z=1 TO 32
9160 LET A(VAL B$( TO 2))=B
9170 LET B$=B$(3 TO )
9180 NEXT Z
9190 FOR Z=1 TO 14
9200 LET A(VAL E$( TO 2))=E
9210 LET E$=E$(3 TO )
```

```
9220 NEXT Z
9230 LET COMP=0
9240 LET HUM=0
9400 PRINT AT 0,0;"█ 1 2 3 4 5
6 7 8 ██";AT 1,1;"████████████████
██";AT 18,0;"███████████████████
█";AT 19,0;"██ 1 2 3 4 5 6 7 8 █
"
9410 FOR Z=1 TO 17
9420 PRINT AT Z,1;"█";AT Z,18;"█
"
9430 IF 2*INT (Z/2)=Z THEN PRINT
 AT Z,0;CHR$ (156+Z/2);AT Z,19;C
HR$ (156+Z/2)
9440 IF 2*INT (Z/2)<>Z THEN PRIN
T AT Z,0;"█";AT Z,19;"█"
9450 NEXT Z
9500 RETURN
```



Now here is *Chopper Checkers*:

```
  20 GOSUB 9000
  30 GOSUB 8000
  35 SLOW
  40 GOSUB 7000
  50 GOSUB 6000
  60 IF HUM=7 THEN PRINT AT 19,0
;"YOU WIN ";W
  70 IF COMP=7 THEN PRINT AT 19,
0;"I WIN ";W
  80 GOTO 40
6000 REM **COMPUTER MOVE**
6010 FOR Z=88 TO 11 STEP -1
6020 IF A(Z)=C THEN GOTO 6050
6030 NEXT Z
6040 GOTO 6200
6050 LET Y=-11
6055 IF Z+Y>88 OR Z+Y<11 OR Z+2*
Y>88 OR Z+2*Y<11 THEN GOTO 6070
6060 IF A(Z+Y)=H AND A(Z+2*Y)=E
THEN GOTO 6100
6070 LET Y=-9*(Y=-11)+9*(Y=-9)+1
0*(Y=9)+(Y=100)
6080 IF Y<>0 THEN GOTO 6055
6085 NEXT Z
6090 GOTO 6200
6100 LET A(Z)=E
6110 LET A(Z+Y)=E
6120 LET A(Z+2*Y)=C
6130 LET COMP=COMP+1
6140 PRINT AT 2,22;"COMP.  ";COM
P
6150 LET F=INT ((Z+Y)/10)
6160 LET G=Z+Y-10*F
6170 PRINT AT 2*G,2*F;"  ";AT 2*
G+1,2*F;"  "
6172 LET F=INT ((Z+2*Y)/10)
6174 LET G=Z+2*Y-10*F
6176 PRINT AT 2*G,2*F;"▚▞";AT 2*
G+1,2*F;"▞▚"
6180 LET F=INT (Z/10)
6182 LET G=Z-10*F
6184 PRINT AT 2*G,2*F;"  ";AT 2*
G+1,2*F;"  "
6190 RETURN
6200 REM **NON-CAPTURE MOVE**
6210 FOR Z=1 TO 200
```

```
6220 LET K=INT (RND*78)+11
6230 IF A(K)=C THEN GOTO 6260
6240 NEXT Z
6250 GOTO 6500
6260 LET Y=-11
6280 IF A(K+Y)=E THEN GOTO 6330
6290 LET Y=-9*(Y=11)+9*(Y=-9)+11
*(Y=9)+(Y=100)
6300 IF Y<>0 THEN GOTO 6270
6310 NEXT Z
6320 GOTO 6500
6330 IF K+2*Y>88 OR K+2*Y<11 THE
N GOTO 6400
6340 IF A(K+2*Y)=H THEN GOTO 624
0
6350 IF K-2*Y<11 OR K-2*Y>88 THE
N GOTO 6400
6360 IF A(K-2*Y)=H THEN GOTO 624
0
6400 LET A(K+Y)=C
6410 LET A(K)=E
6420 LET F=INT ((K+Y)/10)
6430 LET G=K+Y-10*F
6440 PRINT AT 2*G,2*F;"▚▞";AT 2*
G+1,2*F;"▞▚"
6450 LET F=INT (K/10)
6460 LET G=K-10*F
6470 PRINT AT 2*G,2*F;"  ";AT 2*
G+1,2*F;"  "
6480 RETURN
6500 FOR G=1 TO 200
6510 LET K=INT (RND*78)+11
6520 IF A(K)=C THEN GOTO 6600
6530 NEXT G
6540 PRINT AT 0,0;"I CONCEDE THE
 GAME"
6550 STOP
6600 IF A(K-11)=E THEN LET Y=-11
6610 IF A(K-11)=E THEN GOTO 6400
6620 IF A(K-9)=E THEN LET Y=-9
6630 IF A(K-9)=E THEN GOTO 6400
6640 GOTO 6540
7000 REM **PLAYER MOVE**
7010 PRINT AT 20,0;"ENTER YOUR M
OVE AS ""3344"""
7020 INPUT A$
7030 IF LEN A$<>4 THEN GOTO 7020
7040 PRINT AT 20,0;"
                    "
7050 LET A1=VAL A$(1)
```

```
7055 LET A2=VAL A$(2)
7060 LET B1=VAL A$(3)
7065 LET B2=VAL A$(4)
7070 LET A(10*B1+B2)=H
7080 LET A(10*A1+A2)=E
7090 PRINT AT 2*B2,2*B1;"▟▙";AT
2*B2+1,2*B1;"▜▛"
7100 PRINT AT 2*A2,2*A1;"  ";AT
2*A2+1,2*A1;"  "
7110 IF ABS (A1-B1)=1 THEN RETUR
N
7120 LET HUM=HUM+1
7125 LET A((10*B1+B2+10*A1+A2)/2
)=E
7130 PRINT AT (A2+B2),(A1+B1);"
  ";AT A2+B2+1,A1+B1;"  "
7140 PRINT AT 0,22;"HUMAN: ";HUM
7990 RETURN
8000 REM PRINT BOARD - START
8040 FOR Z=8 TO 1 STEP -1
8060 FOR X=1 TO 8
8070 IF A(10*Z+X)=H THEN PRINT A
T 2*X,2*Z;"▟▙";AT 2*X+1,2*Z;"▜▛"
8080 IF A(10*Z+X)=C THEN PRINT A
T 2*X,2*Z;"▚▞";AT 2*X+1,2*Z;"▞▚"
8090 IF A(10*Z+X)=B THEN PRINT A
T 2*X,2*Z;"▒▒";AT 2*X+1,2*Z;"▒▒"
8110 NEXT X
8120 NEXT Z
8130 RETURN
8990 STOP
8999 STOP
9000 FAST
9005 DIM A(100)
9010 LET H$="1113151722242628313
33537"
9020 LET C$="8284868871737577626
46668"
9030 LET B$="1214161821232527323
4363841434547525456586163656772
747678818385 87"
9040 LET E$="5153555842444648"
9050 FOR Z=1 TO 100
9060 LET A(Z)=9
9070 NEXT Z
9080 LET H=CODE "H"
9090 LET C=CODE "C"
9095 LET B=CODE "█"
9100 LET E=CODE " "
9105 FOR Z=1 TO 12
```

```
9110 LET A(VAL H$( TO 2))=H
9120 LET A(VAL C$( TO 2))=C
9130 LET H$=H$(3 TO )
9135 LET C$=C$(3 TO )
9140 NEXT Z
9150 FOR Z=1 TO 32
9160 LET A(VAL B$( TO 2))=B
9170 LET B$=B$(3 TO )
9180 NEXT Z
9190 FOR Z=1 TO 8
9200 LET A(VAL E$( TO 2))=E
9210 LET E$=E$(3 TO )
9220 NEXT Z
9230 LET COMP=0
9240 LET HUM=0
9400 PRINT AT 0,0;"███1█2█3█4█5█
6█7█8██";AT 1,1;"███████████████
██";AT 18,0;"███████████████████
█";AT 19,0;"████1█2█3█4█5█6█7█8█
"
9410 FOR Z=1 TO 17
9420 PRINT AT Z,1;"█";AT Z,18;"█
"
9430 IF 2*INT (Z/2)=Z THEN PRINT
 AT Z,0;CHR$ (156+Z/2);AT Z,19;C
HR$ (156+Z/2)
9440 IF 2*INT (Z/2)<>Z THEN PRIN
T AT Z,0;"█";AT Z,19;"█"
9450 NEXT Z
9500 RETURN
```

Card Games

BLACKJACK

Gwyn Dewey's adaptation of the famous casino card game will have you battling to defeat the computer. Very clear prompts are included in this program, which even draws the backs of the cards as they are dealt before turning them over. The aim of *Blackjack*, as you



probably know, is to get a total of 21, or as close to 21 as you can, without exceeding 21. Cards are worth their face value (with picture cards counting as 10), except for the ace, which can be 11 or 1 (if counting it as 11 would cause you to "bust," or exceed 21). You enter **T** to be hit (ask for a new card) or **S** to stick (stay with the hand you have). If you like programs which maximize the graphics potential of the T/S 1000, you'll enjoy watching *Blackjack* in action.

```
  0 REM B.DEWEY BLACKJACK
        15 JUNE 1982
  1 SAVE "BLACKJACK"
  2 LET BET=100
  2 GOSUB 8000
  3 LET B$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒"
  4 PRINT B$
  5 FOR I=1 TO 10
  6 PRINT "▒███████████████████
███████████▒"
  7 PRINT "▒███████████████████
███████████▒"
  8 NEXT I
  9 PRINT B$
 10 RAND
 11 LET V=1
 12 DIM D(17)
 20 PRINT AT 5,13;"BLACKJACK"
 22 LET Z$="1 2 3 4 5 6 7 8 9 A
 B C D 1▀ 2▀ 3▀ 4▀ 5▀ 6▀ 7▀ 8▀ 9▀ A▀ B▀ C▀ D
▀ 1 ▀2 ▀3 ▀4 ▀5 ▀6 ▀7 ▀8 ▀9 ▀A ▀B ▀C ▀D ▀1▀2▀3
▀4▀5▀6▀7▀8▀9▀A▀B▀C▀D▀1▄ 2▄ 3▄ 4▄ 5▄ 6
▄ 7▄ 8▄ 9▄ A▄ B▄ C▄ D▄ "
 23 DIM E$(17,2)
 24 LET F=52
 25 LET Y=0
 26 LET T=0
 27 LET G=0
 28 LET U=0
 29 LET H=0
 30 PRINT AT 8,13;"D=DEAL";AT 1
0,13;"T=HIT ME";AT 12,13;"S=STICK
"
 40 IF INKEY$<>"D" THEN GOTO 40
```

```
  50 PRINT AT 5,13;"████████"
  52 PRINT AT 8,13;"█████";AT 1
0,13;"████████";AT 12,13;"██████
█"
  60 FOR Z=3 TO 13 STEP 10
  65 FOR H=3 TO 23 STEP 5
  67 PRINT AT Z,H;"▌▄▄▐"
  68 PRINT AT Z+1,H;"▌▀▄▀▐"
  69 PRINT AT Z+2,H;"▌█▀█▐"
  70 PRINT AT Z+3,H;"▌█▄█▐"
  71 PRINT AT Z+4,H;"▌▄▀▄▐"
  72 PRINT AT Z+5,H;"▌▀▀▀▐"
  73 NEXT H
  75 NEXT Z
  90 FOR I=1 TO 2
  91 IF I=1 THEN LET H=3
  92 IF I=1 OR I=2 THEN LET Z=12
  93 IF I=2 THEN GOSUB 9000
  94 IF BET<=0 THEN GOTO 9100
  95 IF I=2 THEN LET H=8
  96 LET D(I)=((INT (RND*F))*2)+
1
  97 FOR N=1 TO 6
  98 PRINT AT Z+N,H;A$(((CODE Z$
(D(I)))-28)+((CODE Z$(D(I)+1)*13
)),N)
  99 NEXT N
 100 LET BJ=0
 102 LET E$(I)=Z$(D(I) TO D(I)+1
)
 103 LET F=F-1
 104 LET Z$=Z$( TO D(I)-1)+Z$(D(
I)+2 TO )
 110 NEXT I
 120 FOR J=1 TO 2
 130 IF E$(J,1)>="2" AND E$(J,1)
<="9" THEN LET T=T+VAL E$(J,1)
 150 IF E$(J,1)>"9" THEN LET T=T
+10
 155 IF E$(J,1)="B" THEN LET BJ=
1
 160 IF E$(J,1)="1" THEN LET G=G
+1
 170 IF E$(J,1)="1" THEN LET T=T
+11
 180 NEXT J
 190 IF T=22 THEN LET T=T-10
 200 IF T=22 THEN LET G=G-1
 215 IF G=1 AND BJ=1 THEN GOTO 5
000
```

```
 217 IF T=21 THEN GOTO 6000
 220 LET V=V+1
 230 PRINT AT 10,4;"<H>IT ME,<S>TICK  "
 235 PRINT AT 11,4;"TOTAL ";T;" "
 240 IF INKEY$="S" THEN GOTO 500
 250 IF INKEY$="H" THEN GOTO 270
 260 GOTO 240
 270 PRINT AT 10,4;"                   "
 310 LET I=I+1
 320 LET D(I)=((INT (RND*F))*2)+1
 325 LET H=H+5
 330 FOR N=1 TO 6
 335 PRINT AT Z+N,H;A$(((CODE Z$(D(I)))-28)+((CODE Z$(D(I)+1)*13)),N)
 340 NEXT N
 350 LET E$(I)=Z$(D(I))
 360 LET Z$=Z$( TO D(I)-1)+Z$(D(I)+2 TO )
 370 IF E$(I,1)>="2" AND E$(I,1)<="9" THEN LET T=T+VAL E$(I,1)
 380 IF E$(I,1)>="A" THEN LET T=T+10
 390 IF E$(I,1)="1" THEN LET G=G+1
 395 LET F=F-1
 400 IF E$(I,1)="1" THEN LET T=T+11
 410 IF T>21 AND G=0 THEN GOTO 6500
 430 IF T>21 AND G>0 THEN LET G=G-1
 434 LET V=V+1
 435 IF T>21 THEN LET T=T-10
 436 IF V=5 THEN GOTO 4000
 437 IF T=21 THEN GOTO 6000
 440 GOTO 230
 500 PRINT AT 12,26;"STUCK"
 502 PRINT AT 13,28;T
 504 PRINT AT 10,4;"COMPUTER           "
 506 PRINT AT 11,10;"  "
 510 DIM E$(5)
 515 LET G=0
 520 FOR I=1 TO 2
 530 IF I=1 THEN LET H=3
```

```
 540 IF I=1 OR I=2 THEN LET Z=2
 550 IF I=2 THEN LET H=8
 560 LET D(I)=((INT (RND*F))*2)+
1
 570 FOR N=1 TO 6
 580 PRINT AT Z+N,H;A$(((CODE Z$
(D(I)))-28)+((CODE Z$(D(I)+1)*13
)),N)
 585 LET F=F-1
 590 NEXT N
 595 LET BJ=0
 600 LET E$(I)=Z$(D(I) TO D(I)+1
)
 610 LET F=F-1
 620 LET Z$=Z$( TO D(I)-1)+Z$(D(
I)+2 TO )
 630 NEXT I
 640 FOR J=1 TO 2
 650 IF E$(J)>="2" AND E$(J)<="9
" THEN LET U=U+VAL E$(J)
 660 IF E$(J)>"9" THEN LET U=U+1
0
 665 IF E$(J)="B" THEN LET BJ=1
 670 IF E$(J)="1" THEN LET G=G+1
 680 IF E$(J)="1" THEN LET U=U+1
1
 690 NEXT J
 700 IF U=22 THEN LET U=U-10
 710 IF U=22 THEN LET G=G-1
 712 IF G=1 AND BJ=1 THEN GOTO 3
000
 715 IF U=21 THEN GOTO 2000
 716 PRINT AT 11,10;U
 720 LET C5=2
 730 IF C5=5 AND U<=21 THEN GOTO
 1500
 740 IF U=21 THEN GOTO 2000
 742 IF U>21 AND G<=0 THEN GOTO
3500
 745 IF U>=T AND V<5 THEN GOTO 7
000
 750 IF V=5 AND C5=5 THEN GOTO 2
000
 775 IF C5=5 AND U>21 THEN GOTO
3500
 780 LET H=H+5
 790 LET C5=C5+1
 800 LET D(I)=((INT (RND*F))*2)+
1
 805 PRINT AT 11,10;U
```

```
 810 FOR N=1 TO 6
 820 PRINT AT Z+N,H;A$(((CODE Z$
(D(I))-28)+((CODE Z$(D(I)+1)*13
)),N)
 830 NEXT N
 840 LET E$(I)=Z$(D(I))
 850 LET Z$=Z$( TO D(I)-1)+Z$(D(
I)+2 TO )
 860 IF E$(I)>="2" AND E$(I)<="9
" THEN LET U=U+VAL E$(I)
 870 IF E$(I)>="A" THEN LET U=U+
10
 880 IF E$(I)="1" THEN LET G=G+1
 890 LET F=F-1
 900 IF E$(I)="1" THEN LET U=U+1
1
 910 LET B=U
 930 IF U>21 AND G>0 THEN LET U=
U-10
 935 IF B>21 AND G>0 THEN LET G=
G-1
 936 LET I=I+1
 940 GOTO 730
1500 PRINT AT 10,4;"OOH A FIVE 
CARDER"
1510 PRINT AT 11,4;"I WIN DOUBLE
"
1520 LET BET=BET-(GOBET*2)
1550 GOTO 9800
2000 PRINT AT 10,4;"TWENTY-ONE"
2010 PRINT AT 11,4;"I WIN      "
2020 LET BET=BET-GOBET
2030 GOTO 1530
3000 PRINT AT 10,4;"OH DEAR BLAC
KJACK"
3010 PRINT AT 11,4;"I WIN TREBLE
"
3020 LET BET=BET-(GOBET*3)
3030 GOTO 9800
3500 PRINT AT 10,4;"CURSES..."
3510 PRINT AT 11,4;"I BUST YOU W
IN"
3520 LET BET=BET+GOBET
3530 GOTO 9800
4000 PRINT AT 10,4;"CURSES..."
4010 PRINT AT 11,4;"FIVE CARDER 
WINS DOUBLE"
4020 LET BET=BET+(GOBET*2)
4030 GOTO 9800
```

```
5000 PRINT AT 10,4;"CURSES..."
5010 PRINT AT 11,4;"BLACKJACK WI
NS TREBLE"
5020 LET BET=BET+(GOBET*3)
5030 GOTO 9800
6000 PRINT AT 10,4;"CURSES..."
6010 PRINT AT 11,4;"TWENTY-ONE Y
OU WIN"
6020 GOTO 3520
6500 PRINT AT 10,4;"YOU BUST"
6510 PRINT AT 11,4;"I WIN      "
6520 GOTO 2020
7000 PRINT AT 10,4;"I BEAT YOU";
U;"█";T;"██"
7010 GOTO 2010
7990 STOP
8000 DIM A$(52,6,5)
8002 FAST
8005 FOR I=1 TO 4
8010 IF I=1 THEN LET F$="*"
8020 IF I=2 THEN LET F$="+"
8030 IF I=3 THEN LET F$="V"
8040 IF I=4 THEN LET F$="?"
8050 LET G$="█ "+F$+" █"
8060 LET H$="█ "+F$+" "+F$+" █"
8070 LET I$="█ "+F$+F$+F$+" █"
8080 LET J$="█   █"
8100 LET A$(13*(I-1)+1,1)="█A   █
"
8110 LET A$(13*(I-1)+1,2)=J$
8120 LET A$(13*(I-1)+1,3)=J$
8130 LET A$(13*(I-1)+1,4)=G$
8140 LET A$(13*(I-1)+1,5)=J$
8150 LET A$(13*(I-1)+1,6)="█   A█
"
8160 LET A$(13*(I-1)+2,1)="█2   █
"
8170 LET A$(13*(I-1)+2,2)=G$
8180 LET A$(13*(I-1)+2,3)=J$
8190 LET A$(13*(I-1)+2,4)=J$
8200 LET A$(13*(I-1)+2,5)=G$
8210 LET A$(13*(I-1)+2,6)="█   2█
"
8220 LET A$(13*(I-1)+3,1)="█3   █
"
8230 LET A$(13*(I-1)+3,2)=G$
8240 LET A$(13*(I-1)+3,3)=G$
8250 LET A$(13*(I-1)+3,4)=J$
8260 LET A$(13*(I-1)+3,5)=G$
```

```
8270 LET A$(13*(I-1)+3,6)="█   3█
::
8280 LET A$(13*(I-1)+4,1)="█4   █
::
8290 LET A$(13*(I-1)+4,2)=H$
8300 LET A$(13*(I-1)+4,3)=J$
8310 LET A$(13*(I-1)+4,4)=J$
8320 LET A$(13*(I-1)+4,5)=H$
8330 LET A$(13*(I-1)+4,6)="█   4█
::
8340 LET A$(13*(I-1)+5,1)="█5   █
::
8350 LET A$(13*(I-1)+5,2)=H$
8360 LET A$(13*(I-1)+5,3)=G$
8370 LET A$(13*(I-1)+5,4)=J$
8380 LET A$(13*(I-1)+5,5)=H$
8390 LET A$(13*(I-1)+5,6)="█   5█
::
8400 LET A$(13*(I-1)+6,1)="█6   █
::
8410 LET A$(13*(I-1)+6,2)=H$
8420 LET A$(13*(I-1)+6,3)=H$
8430 LET A$(13*(I-1)+6,4)=J$
8440 LET A$(13*(I-1)+6,5)=H$
8450 LET A$(13*(I-1)+6,6)="█   6█
::
8460 LET A$(13*(I-1)+7,1)="█7   █
::
8470 LET A$(13*(I-1)+7,2)=H$
8480 LET A$(13*(I-1)+7,3)=H$
8490 LET A$(13*(I-1)+7,4)=G$
8500 LET A$(13*(I-1)+7,5)=H$
8510 LET A$(13*(I-1)+7,6)="█   7█
::
8520 LET A$(13*(I-1)+8,1)="█8   █
::
8530 LET A$(13*(I-1)+8,2)=H$
8540 LET A$(13*(I-1)+8,3)=H$
8550 LET A$(13*(I-1)+8,4)=H$
8560 LET A$(13*(I-1)+8,5)=H$
8570 LET A$(13*(I-1)+8,6)="█   8█
::
8580 LET A$(13*(I-1)+9,1)="█9   █
::
8590 LET A$(13*(I-1)+9,2)=H$
8600 LET A$(13*(I-1)+9,3)=H$
8610 LET A$(13*(I-1)+9,4)=I$
8620 LET A$(13*(I-1)+9,5)=H$
8630 LET A$(13*(I-1)+9,6)="█   9█
::
```

```
8640 LET A$(13*(I-1)+10,1)="▌10
▌"
8650 LET A$(13*(I-1)+10,2)=H$
8660 LET A$(13*(I-1)+10,3)=I$
8670 LET A$(13*(I-1)+10,4)=I$
8680 LET A$(13*(I-1)+10,5)=H$
8690 LET A$(13*(I-1)+10,6)="▌  10
▌"
8700 LET A$(13*(I-1)+11,1)="▌J
▌"
8710 LET A$(13*(I-1)+11,2)="▌"+F
$+"▀▀▌"
8720 LET A$(13*(I-1)+11,3)="▌▄▄▄
▌"
8730 LET A$(13*(I-1)+11,4)="▌▀▀▀
▌"
8740 LET A$(13*(I-1)+11,5)="▌▄▄"
+F$+" ▌"
8750 LET A$(13*(I-1)+11,6)="▌   J
▌"
8760 LET A$(13*(I-1)+12,1)="▌Q
▌"
8770 LET A$(13*(I-1)+12,2)="▌"+F
$+"▀▀▌"
8780 LET A$(13*(I-1)+12,3)="▌█▄█
▌"
8790 LET A$(13*(I-1)+12,4)="▌█▀█
▌"
8800 LET A$(13*(I-1)+12,5)="▌▄▄"
+F$+" ▌"
8810 LET A$(13*(I-1)+12,6)="▌   Q
▌"
8815 LET A$(13*(I-1)+13,2)=""
8820 LET A$(13*(I-1)+13,1)="▌K
▌"
8830 LET A$(13*(I-1)+13,2)="▌"+F
$+"▀▀▌"
8840 LET A$(13*(I-1)+13,3)="▌██
▌"
8850 LET A$(13*(I-1)+13,4)="▌ ██
▌"
8860 LET A$(13*(I-1)+13,5)="▌▄▄"
+F$+" ▌"
8870 LET A$(13*(I-1)+13,6)="▌   K
▌"
8880 NEXT I
8887 SLOW
8888 RETURN
9000 PRINT AT 11,4;"BET?"
9010 PRINT AT 11,24;"$";
```

```
9020 LET L$=STR$ BET
9030 FOR Y=1 TO LEN L$
9040 LET L$(Y)=CHR$ ((CODE L$(Y)
)+128)
9050 NEXT Y
9060 PRINT L$
9070 INPUT GOBET
9080 IF GOBET<1 OR GOBET>BET THE
N GOTO 9070
9085 PRINT AT 11,4;"    "
9088 PRINT AT 20,1;"BET $";GOBET
9090 RETURN
9100 CLS
9110 PRINT "YOU HAVE NO MONEY LE
FT.","MY PORTER THREW YOU OUT.",
"YOU ARE IN THE STREET STARVING.
";"PLAY AGAIN?"
9120 INPUT D$
9125 CLS
9130 IF D$="Y" THEN RUN 2
9140 STOP
9800 FOR M=1 TO 50
9810 NEXT M
9820 FOR I=2 TO 20
9830 PRINT AT I,1;"             
                 "
9840 NEXT I
9850 GOTO 10
```

POKER

Hans Meier of Rustenburg, South Africa, has produced a highly challenging version of the card game poker for the T/S 1000. As the program is about 14K long, I'd suggest you SAVE it every so often as you are entering the program, so that if you lose what you've entered, all your work is not wasted.

The value of each card is the character set number of the first character. For example, a 5D is a five of diamonds and its value is 33 (the CHR$ value of 5), and similarly AE is a ten of hearts and its value is 38.

The whole pack of cards is stored in a string, A$, line 2050.

The actual program begins at line 2040 where A$ and various variables are assigned. The arrays are also dimensioned here. Control then goes to the subroutine beginning in line 10. A$ is then randomly "split" in line 70 and the two parts then put together "back-to-front" into a string (X$) in line 80.

Lines 90–130 make up loop in which 18 cards (C$) are selected from this string at random. The "card" selected is removed from the string in line 120 so that it cannot be "dealt" again. After printing the card outlines (GOSUB 430), control goes back to the main pro-

gram where the hands are dealt in lines 2190–2240. The hands are kept in two string arrays, M$(1), M$(2) and Y$(1), Y$(2), which represent the computer and human strings respectively.

Line 450 (with all shifts in the GRAPHICS mode) reads as follows:

450 PRINT AT A,B;"shift E + 3*shift 7 + shift R"; **AT A+1,B;**"shift 5 + 3 spaces + shift 8"; **AT A + 2,B;**
"ditto"; **AT A+3,B;**"ditto"; **AT A+4,B;**"ditto";
AT A+5,B;"ditto"; **AT A+5,B;** "shift W + 3*shift 6 + shift Q"

The hands are first subjected to a bubble sort. To do this, control goes to the subroutine in line 200. The value used to sort the cards is the character set number of the first character of each card. On returning to the main program, this sorted string is once again placed in M$(1) or Y$(1). The first version is used for evaluation and manipulation of the hand and the second is used for printing the hands to the display.

Now the control then goes to the subroutine in line 320 where the values of the cards in the hands are changed to real card values. This subroutine consists of a loop which extracts one card (a two-character string), at a time. If the first part of a string is a A,B,C,D, or E it is changed to a T(Ten), J(Jack), Q(Queen), K(King), or A(Ace) respectively. If the second letter is an E or F it is changed to an H or S respectively. (CE would become QH.) On return to the main program this string is loaded into M$(2) or Y$(2).

When you have discarded the cards that you do not require (lines 2380–2510), your hand is once again subjected to a sort (GOSUB 200) and change (GOSUB 320). Control then goes to the subroutine in line 480 where the value of the hand is worked out. This is done by allocating an initial value to the hand, then making comparisons to determine whether such things as pairs

or threes exist, and adding another number to the initial one to obtain a final value.

The initial value of a hand is as follows: 0 for a high card, 50 for a pair, 100 for two pairs, 150 for threes, 200 for a straight, 250 for a flush, 300 for a full house, 350 for four of a kind, 400 for a straight flush, and 450 for a royal flush. To this value is added the "value" of the highest card or pair, etc. For example, in line 560 the computer searches for a "full house." If one exists the control goes to line 810, where the initial value for a full house (V=300) is allocated. Then a search is made for the threes in the hand, and their character set value is added to the 300. This enables the computer to determine which hand wins by comparison of MV with YV.

Now comes the computer's turn. Its hand is subjected to the same sort and evaluation. Control then goes to the subroutine in line 900. Right at the start of this subroutine the variable T (the number of cards taken) is 0. The computer decides from the "value" of its hand (MV) which line within the subroutine it will go to. Should the "value" of the hand be greater than 200 (a straight) then T=0 and control returns to the main program. Depending on what the hand is, the computer now decides on how many cards to discard. If it has a pair it goes from line 950 (MV>50 but not >100) to line 1330, where the computer searches for the pair and then replaces the other three cards with cards from the pack (C$). On returning to the main program the value T is used to print the statement **DEALER TAKES** [T] **CARDS** to the screen.

The computer again sorts its hand, changes M$(1) for printing, and determines the value of its new hand (lines 2690–2780). The computer's card outlines are also printed (line 2750).

The betting begins in line 2790. When you have placed your bet and it is "legal," i.e. not higher than

the limit or than your credit, control goes to the subroutine in line 1550. If you enter a 0 for your initial bet the ZX81 presumes you want to throw in your cards, and starts a new deal.

Depending on the value of its hand, the computer decides how far it will raise you and whether it will see you or throw in its cards. But beware—there is also a random "bluff" factor built in. Whichever way the betting goes the subroutine ultimately returns control to the main program at line 2930.

On returning, the computer prints its hand to the screen, unless you have thrown in your cards. Control then passes to the subroutine in line 1900 where the hand is described (V$). This happens twice, once for your hand and once for the computer's. On each return the description of the hand is printed under the appropriate hand, e.g. "You have... V$...". In line 3040 the computer finally decides who wins and prints the necessary comment to the screen.

Between lines 3050 and 3130 the score is adjusted and a check is made to see whether you have no more money or whether you have broken the bank. The control then passes to line 2180, where we start a new game.

There is one other subroutine that I have not mentioned, the one on line 1940. This is merely a delay loop and helps to "stall" while information is on the screen.

Beginning at line 3140 are a few comments that are necessary during the game. The instructions are contained from line 3310 onward.

If the program is too slow for your liking it is possible to speed it up considerably by letting the computer switch from SLOW to FAST and vice versa during the running of the program. If you want to use the FAST option I would suggest inserting the following lines.

```
2175 FAST
```

2375 SLOW
2515 FAST
2785 SLOW.

PLAYING THE GAME

To start the program use the command RUN 2040. After asking whether you require instructions or not, the computer "shuffles" the pack and tells you it's ready: **READY PRESS N/L**. After a slight pause your hand is displayed on the screen. You now have the choice of "throwing out" some of your cards (up to four). When you have entered the numbers of the cards you want to discard, press **P** for pass to continue. Your new hand will now appear on the screen. You will be asked to bet. If you want to throw in your cards enter **0**. To see the computer, enter the amount that you bet. To raise, enter twice that amount.

For the rest the program is pretty self-explanatory. I hope you enjoy the game as much as I do. Good luck.

THE VARIABLES

A$—The initial pack of cards

C$—The random sequence of 18 cards from which the hands are dealt

E$—A string used in the bubble sort routine to store the result

K$—A string used to take and return various strings to and from subroutines

Q$—A string consisting of 32 spaces which is printed over certain lines on the screen when it is necessary to clear them and not the whole screen

X$—The abridged pack from which the card is dealt, and which is then updated

V$—The string describing the type of hand

M$(1)—The computer's hand in its uncorrected state (except for sorting)

Y$(1)—Ditto the player

M$(2)—The computer's hand consisting of 10 characters, each pair depicting a card
Y$(2)—Ditto the player
A—Used to tell the computer where to begin printing the card outlines
B—Used to tell the computer where to begin printing the card values
I—The value of the individual bets during the game. Can either be = 0, = BTG (player sees computer) or = 2*BTG(player raises)
J—Used to carry the value of a hand to the subroutine containing the strings describing the hands
T—The number of cards discarded by the computer
V—The value of a hand as determined in the subroutine starting at line 540
Y—The player's initial credit
MV—The value of the computer's hand
YV—Ditto the player
WL—Used to carry the amount placed on the table for the deal to various routines in order to update your credit
BTC—The original bet by the player.

```
  0 *** H.O.MEIER MARCH 82 ***
 10 REM SHUFFLE CARDS
 20 LET C$=""
 30 PRINT AT 20,0;Q$+Q$;AT 21,0
;"STAND BY PLEASE"
 40 GOSUB 2020
 50 RAND
 60 LET C$=""
 70 LET R=INT (RND*52)*2+1
 80 LET X$=A$(R TO )+A$( TO R-1
)
 90 FOR L=1 TO 18
100 LET R=INT (RND*(LEN X$/2))*
2+1
110 LET C$=C$+X$(R TO R+1)
120 LET X$=X$( TO R-1)+X$(R+2 T
O )
```

```
 130 NEXT L
 140 PRINT AT 21,0;"READY  PRESS
 ENTER"
 150 INPUT R$
 160 CLS
 170 LET A=0
 180 GOSUB 0430
 190 RETURN
 200 REM SORT HI-LO
 210 LET R=1
 220 FOR L=1 TO 7 STEP 2
 230 IF CODE K$(L)<CODE K$(L+2)
THEN GOSUB 270
 240 NEXT L
 250 IF R=0 THEN GOTO 200
 260 RETURN
 270 LET E$=K$(L TO L+1)
 280 LET K$(L TO L+1)=K$(L+2 TO
L+3)
 290 LET K$(L+2 TO L+3)=E$
 300 LET R=0
 310 RETURN
 320 REM CHANGE CARD VALUES
 330 FOR L=1 TO 9 STEP 2
 340 IF K$(L)="A" THEN LET K$(L)
="T"
 350 IF K$(L)="B" THEN LET K$(L)
="J"
 360 IF K$(L)="C" THEN LET K$(L)
="0"
 370 IF K$(L)="D" THEN LET K$(L)
="K"
 380 IF K$(L)="E" THEN LET K$(L)
="A"
 390 IF K$(L+1)="E" THEN LET K$(
L+1)="H"
 400 IF K$(L+1)="F" THEN LET K$(
L+1)="S"
 410 NEXT L
 420 RETURN
 430 REM CARD OUTLINE
 440 FOR B=1 TO 25 STEP 6
 450 PRINT AT A,B;"█▀▀▀▀█";AT A+1
,B;"█    █";AT A+2,B;"█    █";AT A
+3,B;"█    █";AT A+4,B;"█    █";AT
 A+5,B;"█    █";AT A+6,B;"█▄▄▄▄█"
 460 NEXT B
 470 RETURN
 480 REM CARD VALUES
 490 FOR A=2 TO 26 STEP 6
```

```
 500 PRINT AT B,A;K$(1);AT B+2,A
+1;K$(2);AT B+4,A+2;K$(1)
 510 LET K$=K$(3 TO )
 520 NEXT A
 530 RETURN
 540 REM DETERMINE HAND VALUES
 550 LET V=CODE K$
 560 IF K$(1)=K$(3) AND K$(1)=K$
(5) AND K$(7)=K$(9) OR K$(1)=K$(
3) AND K$(5)=K$(7) AND K$(5)=K$(
9) THEN GOTO 0810
 570 IF K$(1)=K$(3) AND K$(1)=K$
(5) AND K$(1)=K$(7) OR K$(3)=K$(
5) AND K$(3)=K$(7) AND K$(3)=K$(
9) THEN GOTO 0860
 580 IF K$(1)=K$(3) AND K$(1)=K$
(5) OR K$(3)=K$(5) AND K$(3)=K$(
7) OR K$(5)=K$(7) AND K$(5)=K$(9
) THEN GOTO 0760
 590 IF K$(1)=K$(3) AND K$(5)=K$
(7) OR K$(1)=K$(3) AND K$(7)=K$(
9) OR K$(3)=K$(5) AND K$(7)=K$(9
) THEN GOTO 0710
 600 IF K$(1)=K$(3) OR K$(3)=K$(
5) OR K$(5)=K$(7) OR K$(7)=K$(9)
 THEN GOTO 0660
 610 IF CODE K$(1)=CODE K$(3)+1
AND CODE K$(3)=CODE K$(5)+1 AND
CODE K$(5)=CODE K$(7)+1 AND CODE
 K$(7)=CODE K$(9)+1 THEN LET V=2
00+CODE K$
 620 IF CODE K$(2)=CODE K$(4) AN
D CODE K$(4)=CODE K$(6) AND CODE
 K$(6)=CODE K$(8) AND CODE K$(8)
=CODE K$(10) THEN LET V=250+CODE
 K$
 630 IF K$(2)=K$(4) AND K$(4)=K$
(6) AND K$(6)=K$(8) AND K$(8)=K$
(10) AND CODE K$(1)=CODE K$(3)+1
 AND CODE K$(3)=CODE K$(5)+1 AND
 CODE K$(5)=CODE K$(7)+1 AND COD
E K$(7)=CODE K$(9)+1 THEN LET V=
400+CODE K$
 640 IF K$(2)=K$(4) AND K$(4)=K$
(6) AND K$(6)=K$(8) AND K$(8)=K$
(10) AND CODE K$=CODE K$(3)+1 AN
D CODE K$(3)=CODE K$(5)+1 AND CO
DE K$(5)=CODE K$(7)+1 AND CODE K
$(7)=CODE K$(9)+1 AND CODE K$=42
 THEN LET V=450+CODE K$
```

```
 650 RETURN
 660 FOR L=1 TO 7 STEP 2
 670 IF K$(L)=K$(L+2) THEN GOTO
0690
 680 NEXT L
 690 LET V=50+CODE K$(L)
 700 RETURN
 710 FOR L=1 TO 3 STEP 2
 720 IF K$(L)=K$(L+2) THEN GOTO
0740
 730 NEXT L
 740 LET V=100+CODE K$(L)
 750 RETURN
 760 FOR L=1 TO 5 STEP 2
 770 IF K$(L)=K$(L+2) THEN GOTO
0790
 780 NEXT L
 790 LET V=150+CODE K$(L)
 800 RETURN
 810 FOR L=1 TO 5 STEP 4
 820 IF K$(L)=K$(L+2) AND K$(L)=
K$(L+4) THEN GOTO 0840
 830 NEXT L
 840 LET V=300+CODE K$(L)
 850 RETURN
 860 FOR L=1 TO 3 STEP 2
 870 IF K$(L)=K$(L+2) THEN LET V
=350+CODE K$(L)
 880 NEXT L
 890 RETURN
 900 REM  TIMEX/SINCLAIR DECIDES
 ON HOW MANY CARDS TO DRAW
 910 LET O$=K$
 920 LET T=0
 930 IF V>200 THEN RETURN
 940 IF V>100 THEN GOTO 1190
 950 IF V>50 THEN GOTO 1330
 960 IF K$(2)=K$(4) AND K$(2)=K$
(6) AND K$(2)=K$(8) OR K$(2)=K$(
4) AND K$(2)=K$(6) AND K$(2)=K$(
10) OR K$(2)=K$(4) AND K$(2)=K$(
8) AND K$(2)=K$(10) OR K$(4)=K$(
6) AND K$(4)=K$(8) AND K$(4)=K$(
10) THEN GOTO 1420
 970 IF K$(2)=K$(4) AND K$(2)=K$
(6) OR K$(2)=K$(4) AND K$(2)=K$(
8) OR K$(2)=K$(4) AND K$(2)=K$(1
0) OR K$(2)=K$(6) AND K$(2)=K$(8
) OR K$(2)=K$(4) AND K$(2)=K$(10
) OR K$(2)=K$(6) AND K$(2)=K$(10
```

```
) OR K$(4)=K$(6) AND K$(4)=K$(8)
 OR K$(4)=K$(8) AND K$(4)=K$(10)
 OR K$(6)=K$(8) AND K$(6)=K$(10)
 OR K$(6)=K$(4) AND K$(6)=K$(10)
 THEN GOTO 1020
 980 IF CODE K$(1)=CODE K$(3)+1
AND CODE K$(3)=CODE K$(5)+1 OR C
ODE K$(3)=CODE K$(5)+1 AND CODE
K$(5)=CODE K$(7)+1 OR CODE K$(5)
=CODE K$(7)+1 AND CODE K$(7)=COD
E K$(9)+1 THEN GOTO 1110
 990 LET K$(3 TO )=C$( TO 8)
1000 LET T=4
1010 RETURN
1020 IF K$(2)=K$(4) AND K$(2)=K$
(6) OR K$(2)=K$(4) AND K$(2)=K$(
8) OR K$(2)=K$(4) AND K$(2)=K$(1
0) OR K$(2)=K$(6) AND K$(2)=K$(1
0) OR K$(2)=K$(6) AND K$(2)=K$(8
) OR K$(2)=K$(10) THEN LET G$=K$
(2)
1030 IF K$(4)=K$(6) AND K$(4)=K$
(8) OR K$(4)=K$(6) AND K$(4)=K$(
10) OR K$(4)=K$(8) AND K$(4)=K$(
10) THEN LET G$=K$(4)
1040 IF K$(6)=K$(8) AND K$(6)=K$
(10) OR K$(6)=K$(8) AND K$(6)=K$
(10) THEN LET G$=K$(6)
1050 FOR L=2 TO 10 STEP 2
1060 IF K$(L)<>G$ THEN LET K$(L-
1 TO L)=C$( TO 2)
1070 LET C$=C$(3 TO )
1080 NEXT L
1090 LET T=2
1100 RETURN
1110 IF CODE K$(1)=CODE K$(3)+1
AND CODE K$(3)=CODE K$(5)+1 AND
CODE K$(5)=CODE K$(7)+1 OR CODE
K$(3)=CODE K$(5)+1 AND CODE K$(5
)=CODE K$(7)+1 AND CODE K$(7)=CO
DE K$(9)+1 THEN GOTO 1500
1120 IF CODE K$(1)=CODE K$(3)+1
THEN LET O$(7 TO 10)=C$( TO 4)
1130 IF CODE K$(3)=CODE K$(5)+1
AND CODE K$(5)=CODE K$(7)+1 THEN
 LET O$(9 TO 10)=C$(3 TO 4)
1140 IF CODE K$(3)=CODE K$(5)+1
AND CODE K$(5)=CODE K$(7)+1 THEN
 LET O$( TO 2)=C$( TO 2)
1150 IF CODE K$(7)=CODE K$(9)+1
```

```
THEN LET O$( TO 4)=C$( TO 4)
1160 LET T=2
1170 LET K$=O$
1180 RETURN
1190 IF V>150 THEN GOTO 1260
1200 IF K$(1)=K$(3) AND K$(5)=K$
(7) THEN LET O$(9 TO )=C$( TO 2)
1210 IF K$(1)=K$(3) AND K$(7)=K$
(9) THEN LET O$(5 TO 6)=C$( TO 2
)
1220 IF K$(3)=K$(5) AND K$(7)=K$
(9) THEN LET O$( TO 2)=C$( TO 2)
1230 LET T=1
1240 LET K$=O$
1250 RETURN
1260 IF K$(1)=K$(3) AND K$(1)=K$
(5) THEN LET O$(7 TO )=C$( TO 4)
1270 IF K$(3)=K$(5) AND K$(3)=K$
(7) THEN LET O$( TO 2)=C$( TO 2)
1280 IF K$(3)=K$(5) AND K$(3)=K$
(7) THEN LET O$(9 TO )=C$(3 TO 4
)
1290 IF K$(5)=K$(7) AND K$(5)=K$
(9) THEN LET O$( TO 4)=C$( TO 4)
1300 LET T=2
1310 LET K$=O$
1320 RETURN
1330 IF K$(1)=K$(3) THEN LET O$(
5 TO )=C$( TO 6)
1340 IF K$(3)=K$(5) THEN LET O$(
 TO 2)=C$( TO 2)
1350 IF K$(3)=K$(5) THEN LET O$(
7 TO 10)=C$(3 TO 6)
1360 IF K$(5)=K$(7) THEN LET O$(
 TO 4)=C$( TO 4)
1370 IF K$(5)=K$(7) THEN LET O$(
9 TO )=C$(5 TO 6)
1380 IF K$(7)=K$(9) THEN LET O$(
 TO 6)=C$( TO 6)
1390 LET T=3
1400 LET K$=O$
1410 RETURN
1420 IF K$(2)=K$(4) OR K$(2)=K$(
6) THEN LET V$=K$(2)
1430 IF K$(2)<>K$(4) AND K$(4)=K
$(6) THEN LET V$=K$(4)
1440 FOR L=2 TO 10 STEP 2
1450 IF K$(L)<>V$ THEN LET K$(L-
1 TO L)=C$( TO 2)
1460 LET C$=C$(3 TO )
```

```
1470 NEXT L
1480 LET T=1
1490 RETURN
1500 IF CODE K$(1)=CODE K$(3)+1
AND CODE K$(3)=CODE K$(5)+1 AND
CODE K$(5)=CODE K$(7)+1 THEN LET
 O$(9 TO )=C$( TO 2)
1510 IF CODE K$(3)=CODE K$(5)+1
AND CODE K$(5)=CODE K$(7)+1 AND
CODE K$(7)=CODE K$(9)+1 THEN LET
 O$( TO 2)=C$( TO 2)
1520 LET K$=O$
1530 LET T=1
1540 RETURN
1550 REM BETTING ROUTINE
1560 LET C=0
1570 IF MV>200 THEN LET C=MV/50
1580 IF MV>200 THEN GOTO 1680
1590 LET X=RND
1600 IF X>.9 THEN LET C=8
1610 IF C=8 THEN GOTO 1680
1620 IF X>.45 OR BTG<3 THEN LET
C=2
1630 IF C=2 THEN GOTO 1680
1640 LET Y=Y+BTG
1650 LET WL=5
1660 GOSUB 1940
1670 GOTO 3190
1680 FOR L=1 TO C
1690 PRINT AT 19,0;O$;AT 19,0;"I
 SEE YOU AND RAISE YOU $";BTG
1700 PRINT AT 21,0;O$;AT 21,0;"E
NTER YOUR BET"
1710 GOSUB 2020
1720 INPUT I
1730 IF Y<2*BTG OR I=0 OR I=BTG
OR I=2*BTG THEN GOTO 1760
1740 PRINT AT 21,0;"SEE OR RAISE
 ME"
1750 GOTO 1720
1760 LET WL=WL+I
1770 LET Y=Y-I
1780 IF Y<BTG THEN GOTO 1980
1790 IF I=0 THEN GOTO 3150
1800 IF I=BTG THEN GOTO 1880
1810 PRINT AT 19,0;O$;AT 19,0;"S
O YOU SEE ME AND RAISE ME..."
1820 GOSUB 1940
1830 IF MV<100 AND L>3 AND RND>=
.6 THEN GOTO 3170
```

```
1840 NEXT L
1850 GOSUB 1940
1860 PRINT AT 19,0;Q$;AT 19,0;"I
 SEE YOU...."
1870 RETURN
1880 PRINT AT 19,0;Q$;AT 19,0;"S
O YOU SEE ME...."
1890 RETURN
1900 REM MAKE THE HANDS
1910 LET V$=("A HIGH CARD" AND J
<50)+("A PAIR" AND J>50 AND J<99
)+("TWO PAIRS" AND J>100 AND J<1
50)+("THREES" AND J>149 AND J<20
0)+("A STRAIGHT" AND J>199 AND J
<250)+("A FLUSH" AND J>249 AND J
<300)+("A FULL HOUSE" AND J>299
AND J<350)+("FOUR OF A KIND" AND
 J>349 AND J<400)+("A STRAIGHT F
LUSH" AND J>399 AND J<450)+("A R
OYAL FLUSH" AND J>449)
1920 LET PP=(22-LEN V$)/2
1930 RETURN
1940 REM DELAY LOOP
1950 FOR P=1 TO 20
1960 NEXT P
1970 RETURN
1980 GOSUB 1940
1990 PRINT AT 19,0;Q$;AT 19,0;"Y
OU WILL HAVE TO SEE ME...."
2000 GOSUB 1940
2010 RETURN
2020 PRINT AT 21,18;"YOU HAVE $"
;Y
2030 RETURN
2040 REM START OF PROGRAM
2050 LET A$="2C3C4C5C6C7C8C9CACB
CCCDCEC2D3D4D5D6D7D8D9DADBDCDDDE
D2E3E4E5E6E7E8E9EAEBECEDEEE2F3F4
F5F6F7F8F9FAFBFCFDFEF"
2060 LET Q$="
              "
2070 DIM M$(2,10)
2080 DIM Y$(2,10)
2090 LET BT=0
2100 LET Y=250
2110 PRINT AT 9,4;" TIMEX/SINCLA
IR POKER "
2120 REM INSTRUCTIONS ?
2130 PRINT AT 20,0;"DO YOU REQUI
RE DETAILS ?(Y OR N)"
```

```
2140 IF INKEY$<>"" THEN GOTO 214
0
2150 IF INKEY$="" THEN GOTO 2150
2160 LET S$=INKEY$
2170 IF S$="Y" THEN GOSUB 3300
2180 GOSUB 0010
2190 REM DEAL HANDS
2200 FOR L=1 TO 9 STEP 2
2210 LET M$(1,L TO L+1)=C$( TO 2
)
2220 LET Y$(1,L TO L+1)=C$(3 TO
4)
2230 LET C$=C$(5 TO )
2240 NEXT L
2250 LET K$=Y$(1)
2260 GOSUB 0200
2270 LET Y$(1)=K$
2280 GOSUB 0320
2290 LET Y$(2)=K$
2300 REM INITIAL STAKE
2310 LET Y=Y-5
2320 LET B=1
2330 GOSUB 0480
2340 LET K$=M$(1)
2350 REM NUMBERS UNDER CARDS
2360 PRINT AT 7,3;"1";AT 7,9;"2"
;AT 7,15;"3";AT 7,21;"4";AT 7,27
;"5"
2370 GOSUB 0200
2380 REM DISCARD WHICH CARDS ?
2390 PRINT AT 19,0;Q$+Q$;AT 19,0
;"SWAP WHICH CARD?  (IE. PRESS 2
 FOR NO. 2 ETC. AND/OR P TO PASS
)"
2400 FOR L=1 TO 4
2410 IF INKEY$<>"" THEN GOTO 241
0
2420 IF INKEY$="" THEN GOTO 2420
2430 LET T$=INKEY$
2440 IF T$="P" THEN GOTO 2510
2450 IF T$<"1" OR T$>"5" THEN GO
TO 2410
2460 PRINT AT 7,VAL T$*6-3;"*"
2470 LET T=VAL T$*2-1
2480 LET Y$(1,T TO T+1)=C$( TO 2
)
2490 LET C$=C$(3 TO )
2500 NEXT L
2510 PRINT AT 19,0;Q$+Q$+Q$;AT 2
1,10;" DEALING "
```

```
2520 LET K$=Y$(1)
2530 GOSUB 0200
2540 LET Y$(1)=K$
2550 GOSUB 0320
2560 LET Y$(2)=K$
2570 LET B=1
2580 GOSUB 0480
2590 PRINT AT 7,0;O$
2600 PRINT AT 21,0;O$;AT 21,0;"B
ET TILL NOW $5"
2610 GOSUB 2020
2620 LET K$=Y$(1)
2630 GOSUB 0540
2640 LET YV=V
2650 LET K$=M$(1)
2660 GOSUB 0540
2670 GOSUB 0900
2680 PRINT AT 19,0;O$+O$;AT 19,0
;"DEALER TAKES ";T;" CARD"+("S"
AND T<>1)
2690 LET M$(1)=K$
2700 GOSUB 0200
2710 LET M$(1)=K$
2720 GOSUB 0320
2730 LET M$(2)=K$
2740 LET A=10
2750 GOSUB 0430
2760 LET K$=M$(1)
2770 GOSUB 0540
2780 LET MV=V
2790 REM BETTING BEGINS
2800 PRINT AT 21,0;O$;AT 21,0;"Y
OUR BET ?"
2810 GOSUB 2020
2820 INPUT BTG
2830 LET BTG=INT BTG
2840 IF BTG=0 THEN GOTO 3150
2850 IF BTG>Y OR BTG>25 THEN PRI
NT AT 21,0;O$
2860 PRINT AT 21,0;("YOU DO NOT
HAVE THAT MUCH..." AND BTG>Y)+("
LIMIT IS $25" AND BTG>25)
2870 IF BTG>Y OR BTG>25 THEN GOS
UB 2020
2880 IF BTG>Y OR BTG>25 THEN GOT
O 2820
2890 IF BTG=Y OR Y-BTG<BTG THEN
LET S=1
2900 LET Y=Y-BTG
2910 LET WL=BTG+5
```

```
2920 GOSUB 1550
2930 LET K$=M$(2)
2940 LET B=11
2950 GOSUB 0480
2960 LET J=YV
2970 GOSUB 1900
2980 PRINT AT 19,0;Q$+Q$;AT 8,PP
;"YOU HAVE ";V$
2990 LET J=MV
3000 GOSUB 1900
3010 PRINT AT 18,PP;" I HAVE ";V
$
3020 GOSUB 1940
3030 REM WHO WINS
3040 PRINT AT 21,0;Q$;AT 21,0;("
OK..., YOU WIN" AND YV>MV)+("TOU
GH LUCK, I WIN" AND MV>YV)+("IT
IS A DRAW..." AND MV=YV)
3050 REM ADJUST MONEY
3060 IF YV>MV THEN LET Y=Y+WL*2
3070 IF YV=MV THEN LET Y=Y+WL
3080 GOSUB 2020
3090 GOSUB 1940
3100 GOSUB 1940
3110 IF Y>2000 THEN GOTO 3220
3120 IF Y<6 THEN GOTO 3270
3130 GOTO 2180
3140 REM VARIOUS REMARKS
3150 PRINT AT 19,0;Q$+Q$;AT 19,0
;"SO YOUR CHICKEN..... ?"
3160 GOTO 3080
3170 LET WL=WL-I
3180 LET Y=Y+I
3190 PRINT AT 19,0;Q$+Q$;AT 19,0
;"I AM OUT............."
3200 LET Y=Y+WL*2
3210 GOTO 3080
3220 GOSUB 1940
3230 GOSUB 1940
3240 CLS
3250 PRINT AT 10,0;"WELL THAT BE
ATS ME. I AM GOING  HOME. BYE-BY
E FOR NOW."
3260 STOP
3270 CLS
3280 PRINT AT 10,0;"IT APPEARS Y
OU HAVE NO MORE CASHTHANKS AND B
YE-BYE."
3290 STOP
3300 CLS
```

```
3310 PRINT AT 1,2;"THIS IS A GAM
E OF DRAW POKER  PLAYED BY YOU A
GAINST THE       TIMEX/SINCLAIR.
"
3320 PRINT AT 5,2;"YOU WILL BE D
EALT WITH FIVE   CARDS, AND THEN
 BE ASKED WHICH  ONES YOU WISH T
O DISCARD. YOU   MAY DISCARD A M
AXIMUM OF 4 CARDS"
3330 PRINT AT 10,2;"ON EACH DEAL
 YOUR TOTAL WILL  AUTOMATICALLY
BE DEBITED WITH   $5. THIS IS YO
UR STARTING STAKE."
3340 PRINT AT 14,2;"THE MAXIMUM
INITIAL BET IS    $25. THEREAFTE
R YOU MUST ALL-   WAYS DOUBLE OR
 SEE. TO THROW IN YOUR CARDS ENT
ER A 0."
3350 PRINT AT 20,2;"GOOD LUCK...
.."
3355 PAUSE 300
3360 RETURN
```

Brain Games

FLIP

Flip is an intriguing game which provides quite a bit of mental stimulation. You'll see a random mix of asterisks and solid squares on a 3 × 3 grid when you start the game. You have to try to end up with eight asterisks surrounding a black square in the middle of the grid. You can only "flip" (a term to be described shortly) an asterisk. You move by entering the number of the piece you wish to flip. Flipping a corner piece causes those adjoining it to change to their opposites (that is, an asterisk becomes a solid square, and vice versa). Flipping a middle piece on one side changes the two either side of it, and flipping the middle one changes the middle piece on all four sides. The piece you flip also changes.

The number of moves you have taken so far is displayed. The program will pause at the end of a game to tell you how many moves it took you to solve it, and then you'll be given a new starting position.

```
NUMBER OF █ IS 1
YOU NEED ONLY ONE, IN THE
MIDDLE SQUARE (5)
```

```
MOVE NUMBER 10

WHICH ONE TO CHANGE?


     1   2   3     █   █   █

     4   5   6     *   █   █

     7   8   9     *   █   *



NUMBER OF █ IS 6
YOU NEED ONLY ONE, IN THE
MIDDLE SQUARE (5)



YOU SOLVED IT IN 15 MOVES



     1   2   3     *   *   *

     4   5   6     *   █   *

     7   8   9     *   *   *
```

```
   7 LET M=0
   8 DIM A(10)
   9 DIM F(4)
  10 LET O=CODE "*"
  15 LET X=CODE "█"
  20 FOR C=1 TO 9
  30 LET B=INT (RND+.5)
  35 LET A(C)=0
  40 IF B=0 THEN LET A(C)=X
  45 NEXT C
  50 GOSUB 180
  60 LET N=0
  70 FOR C=1 TO 9
  80 IF A(C)=X THEN LET N=N+1
  90 NEXT C
  95 PRINT AT 18,0;"NUMBER OF █
IS █";AT 18,15;N
  96 PRINT "YOU NEED ONLY ONE, I
N THE       MIDDLE SQUARE (5)"
 100 IF N=1 AND A(5)=X THEN GOTO
 270
 110 LET M=M+1
 115 PRINT AT 1,0;"MOVE NUMBER "
;M
 120 PRINT AT 3,0;"WHICH ONE TO
CHANGE?"
 121 IF INKEY$<>"" THEN GOTO 120
 122 LET A$=INKEY$
 123 PRINT AT 3,19;"?"
 124 PRINT AT 3,19;"█"
 126 IF A$="" THEN GOTO 122
 127 LET N=VAL A$
 128 IF N<1 OR N>9 THEN GOTO 125
 129 PRINT AT 3,0;"
                 "
 130 GOSUB 310
 140 GOTO 50
 170 STOP
 180 PRINT AT 8,3;"1  2  3   ";C
HR$ (A(1));"  ";CHR$ (A(2));"  "
;CHR$ (A(3))
 200 PRINT AT 10,3;"4  5  6   ";
CHR$ (A(4));"  ";CHR$ (A(5));"
";CHR$ (A(6))
 210 PRINT AT 12,3;"7  8  9   ";
CHR$ (A(7));"  ";CHR$ (A(8));"
";CHR$ (A(9))
 230 RETURN
```

```
270 PRINT
275 PRINT AT 0,0;"YOU SOLVED IT
IN ";M;" MOVES"
280 FOR T=1 TO 500
290 NEXT T
300 CLS
305 RUN
310 IF A(N)=X THEN RETURN
320 IF N=1 THEN LET F(1)=2
321 IF N=1 THEN LET F(2)=4
322 IF N=1 THEN LET F(3)=5
323 IF N=1 THEN LET F(4)=10
330 IF N=2 THEN LET F(1)=1
331 IF N=2 THEN LET F(2)=3
332 IF N=2 THEN LET F(3)=10
333 IF N=2 THEN LET F(4)=10
340 IF N=3 THEN LET F(1)=2
341 IF N=3 THEN LET F(2)=5
342 IF N=3 THEN LET F(3)=6
343 IF N=3 THEN LET F(4)=10
350 IF N=4 THEN LET F(1)=1
351 IF N=4 THEN LET F(2)=7
352 IF N=4 THEN LET F(3)=10
353 IF N=4 THEN LET F(4)=10
360 IF N=5 THEN LET F(1)=2
361 IF N=5 THEN LET F(2)=4
362 IF N=5 THEN LET F(3)=8
363 IF N=5 THEN LET F(4)=6
370 IF N=6 THEN LET F(1)=3
371 IF N=6 THEN LET F(2)=9
372 IF N=6 THEN LET F(3)=10
373 IF N=6 THEN LET F(4)=10
380 IF N=7 THEN LET F(1)=4
381 IF N=7 THEN LET F(2)=5
382 IF N=7 THEN LET F(3)=8
383 IF N=7 THEN LET F(4)=10
390 IF N=8 THEN LET F(1)=7
391 IF N=8 THEN LET F(2)=9
392 IF N=8 THEN LET F(3)=10
393 IF N=8 THEN LET F(4)=10
400 IF N=9 THEN LET F(1)=8
401 IF N=9 THEN LET F(2)=5
402 IF N=9 THEN LET F(3)=6
403 IF N=9 THEN LET F(4)=10
407 FOR G=1 TO 4
408 LET F=0
410 IF A(F(G))=X THEN LET F=1
420 IF F=1 THEN LET A(F(G))=0
```

```
 425 IF F=0 AND A(F(G))=Q THEN L
ET A(F(G))=X
 430 NEXT G
 440 LET A(N)=X
 450 RETURN
```

CODES

Gwyn Dewey's *Codes* program is a game in which numbers and letters are jumbled up, and you have to try to guess them in sequence. When the prompt appears, you indicate your guess of one of the hidden numbers by entering the number above the chosen gray square. There is a time limit, so do not linger too long in making your decision.

```
   1 PRINT TAB 14;"CODES"
   2 PRINT TAB 7;"(THE FINAL VER
SION)"
   3 PRINT TAB 2;"CODES IS A GAM
E WHERE SO MANY"
   4 PRINT TAB 1;"NUMBERS AND LE
TTERS ARE JUMBLED"
   5 PRINT TAB 1;"UP AND THEN HA
VE TO BE GUESSED"
   6 PRINT TAB 3;"IN SEQUENCE. W
HEN THE PROMPT"
   7 PRINT TAB 2;"APPEARS YOU AR
E TO GUESS ONE"
   8 PRINT TAB 4;"OF THE HIDDEN
NUMBERS BY"
   9 PRINT TAB 2;"ENTERING THE N
UMBER ABOVE THE"
  10 PRINT TAB 1;"CHOSEN GREY SQ
UARE. DO NOT TAKE"
```

```
  11 PRINT TAB 1;"TOO LONG OR TH
E T/S WILL BEAT"
  12 PRINT TAB 2;"YOU (GULP). I
HAVE WARNED YOU."
  13 PRINT TAB 3;"ENTER LEVEL (1
 EASY-16 HARD)"
  14 INPUT L
  15 IF L<1 OR L>16 THEN GOTO 14
  16 PRINT TAB 11;L
  17 PRINT ,,TAB 11;"""GOOD LUCK
"""
  18 FOR Y=1 TO 150
  19 NEXT Y
  20 CLS
  21 LET N$=""
  22 LET A$=""
  23 LET B$=""
  25 FOR X=1 TO L
  28 LET N$=N$+"."
  29 LET A$=A$+CHR$ (X+28)
  30 LET B$=B$+" "
  35 NEXT X
  40 RAND
  50 FOR A=1 TO LEN A$
  60 LET B=INT (RND*LEN A$)+1
  70 LET B$(A)=A$(B)
  80 LET A$=A$( TO B-1)+A$(B+1 T
O )
  90 NEXT A
 100 LET C$=CHR$ 29
 101 PRINT AT 10,0;
 102 FOR X=1 TO L
 103 PRINT "▒ ";
 104 NEXT X
 105 PRINT AT 11,0;
 106 FOR X=1 TO L
 107 PRINT CHR$ (X+28);" ";
 108 NEXT X
 110 INPUT D$
 120 IF CODE D$<29 OR CODE D$>L+
28 THEN GOTO 110
 130 LET C=CODE D$-28
 135 LET N$(C)=B$(C)
 140 IF B$(C)=C$ THEN GOTO 200
 150 PRINT AT 10,(C*2)-2;B$(C)
 160 FOR I=1 TO 50
 170 NEXT I
 180 PRINT AT 10,(C*2)-2;"▒"
 190 GOTO 400
```

```
 200 PRINT AT 10,(C*2)-2;CHR$ ((
CODE C$)+128)
 210 LET C$=CHR$ ((CODE C$)+1)
 220 IF C$=CHR$ (L+29) THEN GOTO
 600
 230 GOTO 110
 400 FAST
 410 FOR I=1 TO L
 420 IF N$(I)=C$ THEN GOTO 550
 430 NEXT I
 440 FOR I=1 TO L
 450 IF N$(I)="." THEN GOTO 465
 460 NEXT I
 465 LET N$(I)=B$(I)
 470 IF B$(I)=C$ THEN GOTO 550
 480 SLOW
 490 PRINT AT 10,(I*2)-2;B$(I)
 500 FOR J=1 TO 50
 510 NEXT J
 520 PRINT AT 10,(I*2)-2;"▒"
 530 GOTO 110
 550 SLOW
 560 PRINT AT 10,(I*2)-2;CHR$ ((
CODE C$)+128)
 570 LET C$=CHR$ ((CODE C$)+1)
 580 IF C$=CHR$ (L+29) THEN GOTO
 700
 585 FOR Z=1 TO 50
 586 NEXT Z
 590 GOTO 400
 600 CLS
 610 PRINT AT 10,11;"CURSES...."
 620 PRINT AT 12,13;"YOU WIN"
 630 PRINT AT 20,1;"PRESS ANY KE
Y TO PLAY AGAIN"
 640 IF INKEY$="" THEN GOTO 640
 645 CLS
 650 GOTO 2
 700 CLS
 710 PRINT AT 10,12;"HA HA..."
 720 PRINT AT 12,14;"I WIN"
 730 GOTO 630
```

2114 BUG

Can you discover the byte in your 2114 chip that has a bug in it before it can get into your program? Can you succeed where others (I!) have failed? Here is a game by Chris Callender to see if you can. A brand new 2114, made in Japan, will appear on the screen, along with instructions when you press **RUN**.

Pressing any key causes your scanner to appear as a white dot on the 2114. Move it around using **5**, **6**, **7**, and **8** (and moving in the direction of the arrows on those keys), and as you do so, the signal on your slightly inaccurate bug-detector will change. If you manage to find the bug, you'll get the message **JUST IN TIME** flashing on the screen. I'll leave it to you to find out what happens if you fail. If—after a few games—you wish to make it harder, change the 83 in line 295 into a smaller number.

```
  1 FAST
 10 FOR A=1 TO 9
 20 PRINT AT A,0;"████████████████████████████████"
 30 NEXT A
 40 FOR A=0 TO 63
```

```
  50 IF A/8=INT (A/8) THEN PLOT
A+2,43
  60 IF A/8=INT (A/8) THEN PLOT
A+2,42
  70 IF A/8=INT (A/8) THEN PLOT
A+2,23
  80 IF A/8=INT (A/8) THEN PLOT
A+2,22
  90 NEXT A
 100 PRINT AT 5,1;"2114 R.A.M. I
S. MADE IN JAPAN"
 110 PRINT AT 11,0;"IN THIS VERY
 2114 LIVES A BUG"
 120 PRINT "(NOT A VERY NICE ONE
 EITHER...)"
 130 PRINT "CAN YOU THE HERO FIN
D IT BEFORE"
 140 PRINT "IT GETS INTO YOUR PR
OGRAM AND"
 150 PRINT "CAUSES A COMPLETE SY
STEM CRASH?"
 160 PRINT "(COMPLETE SYSTEM CRA
SHES ARE"
 170 PRINT "NASTY STUFF...). YOU
 CAN MOVE"
 180 PRINT "YOUR SCANNER AROUND
THE MEMORY"
 190 PRINT "I.C. IF YOU GET ONTO
 THE SAME"
 200 PRINT "BYTE AS THE BUG YOU
GET IT. GOOD"
 210 PRINT "LUCK. HIT ANY KEY."
 220 PAUSE 4E4
 230 FOR A=11 TO 21
 240 PRINT AT A,0;"
                                "
 250 NEXT A
 255 RAND
 260 LET BX=INT (RND*63)
 270 LET BY=41-INT (RND*18)
 280 LET PX=0
 290 LET PY=41
 295 FOR C=1 TO 83
 300 UNPLOT PX,PY
 310 IF PX=BX AND PY=BY THEN GOT
O 1000
 320 PRINT AT 12,0;"SCANNER COOR
DINATES ";PX;",";PY
 330 LET S=1000+INT (RND*2)-(ABS
 (PX-BX)+ABS (PY-BY))
```

```
 340 PRINT AT 13,0;"BUG DETECTOR
 (1-1000)=";S
 350 PRINT " (THIS DETECTOR NEED
S FIXING"
 355 PRINT "      ....IT IS INACCU
RATE)"
 360 PAUSE 4E4
 370 LET A$=INKEY$
 380 PLOT PX,PY
 390 PRINT AT 5,1;"2114 R.A.M. I
.C. MADE IN JAPAN"
 400 IF A$="5" AND PX>0 THEN LET
 PX=PX-1
 410 IF A$="6" AND PY>23 THEN LE
T PY=PY-1
 420 IF A$="7" AND PY<42 THEN LE
T PY=PY+1
 430 IF A$="8" AND PX<63 THEN LE
T PX=PX+1
 440 NEXT C
 450 CLS
 460 PRINT AT 11,10;"TOO LATE"
 470 PAUSE 100
 480 FOR A=1 TO 100
 490 RAND USR 3
 500 NEXT A
 510 STOP
1000 CLS
1010 PRINT AT 11,10;"JUST IN TIM
E"
1020 PAUSE 50
1030 PRINT AT 11,10;"JUST IN TIM
E"
1040 PAUSE 50
1050 GOTO 1010
```

Word and Letter Games

ANAGRAMS

Anagrams, by Ken Mahogany, shows the flexibility of your computer's string handling.

The program asks you to enter a word (such as your first name). The computer then will produce every conceivable combination of the letters in your name. The sample run before the program listing shows some anagrams of the programmer's name.

```
AYOGANMH
YOMNGAHA
AYHGOMAN
MANHYGOA
AOHGAYNM
NOAHMAYG
AAYNGHOM
AGONYAHM
NYOHMAAG
AOHMNAGY
AHGAMYNO
AYMAGNHO
AAONYGHM
OMNHAYGA
GHAAYOMN
AAHYGOMN
GHNAMYOA
YAHMNOGA
```

```
HOYNMAGA
MAYGOAHN
AOGNHYMA
NGOAYAMH
```

```
 10 REM ANAGRAMS
 20 REM (C) K MAHOGANY 1982
 30 PRINT "ENTER YOUR WORD"
 40 INPUT A$
 50 LET N=LEN A$
 55 DIM A(N)
 60 LET A(1)=INT (RND*N)+1
 70 FOR Z=2 TO N
 80 LET A(Z)=INT (RND*N)+1
 90 FOR J=1 TO Z-1
100 IF A(J)=A(Z) THEN GOTO 80
110 NEXT J
120 NEXT Z
130 LET B$=""
140 FOR B=1 TO N
150 LET B$=B$+A$(A(B))
160 NEXT B
170 SCROLL
180 PRINT TAB 4;B$
190 GOTO 60
```

SPECTRAL HANGMAN

This is a fairly straightforward game in which the computer chooses a word from its vocabularly, and then gives you a limited number of guesses (based on the length of the word) to get it right. The vocabulary can easily be changed or extended. *Spectral Hangman* was written by Ken Mahogany.

```
  10 REM SPECTRAL HANGMAN
  20 REM (C)K MAHOGANY 1982
  30 GOSUB 1000
  60 LET N=LEN A$
  70 DIM B(N)
  75 DIM D(N)
  80 FOR G=1 TO N
  90 LET B(G)=CODE A$(G)
 100 LET D(G)=B(G)
 110 NEXT G
 120 FOR J=1 TO N+N/3
 140 GOSUB 410
 150 SCROLL
 160 SCROLL
 170 SCROLL
 180 SCROLL
 190 PRINT "ENTER YOUR GUESS NO.
";J
 200 INPUT C$
```

```
 210 LET F=CODE C$
 220 FOR G=1 TO N
 230 IF D(G)=F THEN LET D(G)=0
 240 NEXT G
 260 NEXT J
 265 GOSUB 410
 270 SCROLL
 275 PRINT "SORRY, TIME IS UP"
 277 SCROLL
 280 GOTO 330
 300 SCROLL
 310 PRINT TAB 8;"WELL DONE"
 315 SCROLL
 320 PRINT "YOU GOT THE WORD IN
";J-1;" GUESSES"
 325 SCROLL
 330 PRINT "THE WORD WAS ";A$
 335 SCROLL
 337 SCROLL
 340 PRINT "PRESS ANY KEY FOR A
NEW GAME"
 345 PAUSE 4E4
 350 FOR G=1 TO 24
 360 SCROLL
 370 NEXT G
 380 RUN
 410 LET H=0
 412 SCROLL
 415 FOR E=1 TO N
 420 IF B(E)=D(E) THEN PRINT "-"
;
 430 IF B(E)<>D(E) THEN PRINT CH
R$ B(E);
 435 IF B(E)<>D(E) THEN LET H=H+
1
 440 NEXT E
 450 IF H=N THEN GOTO 300
 455 SCROLL
 460 PRINT "YOU HAVE GUESSED ";H
;" LETTER";
 470 IF H<>1 THEN PRINT "S"
 480 SCROLL
 490 RETURN
1000 LET K=INT (RND*25+1)*10+150
0
1010 GOSUB K
1020 RETURN
1510 LET A$="FEATURE"
1515 RETURN
1520 LET A$="SPECTRUM"
```

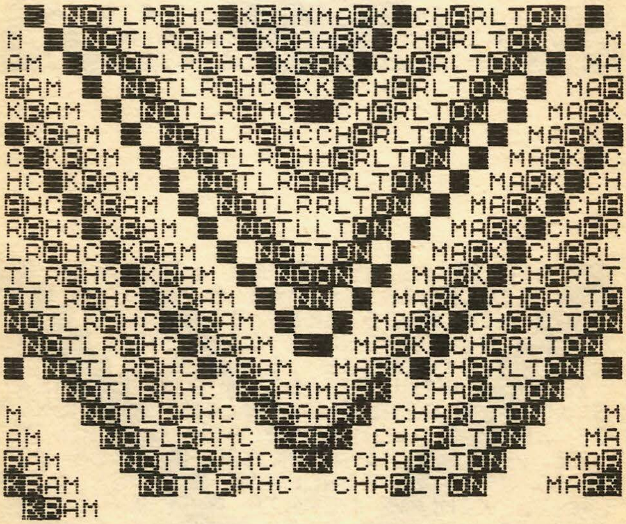
```
1525 RETURN
1530 LET A$="CAMBRIDGE"
1535 RETURN
1540 LET A$="HAZARD"
1545 RETURN
1550 LET A$="PUMPKIN"
1555 RETURN
1560 LET A$="QUESTION"
1565 RETURN
1570 LET A$="QUIZ"
1575 RETURN
1580 LET A$="UNCLE"
1585 RETURN
1590 LET A$="RECORDER"
1595 RETURN
1600 LET A$="BASIC"
1605 RETURN
1610 LET A$="FORMULA"
1615 RETURN
1620 LET A$="FRIENDLY"
1625 RETURN
1630 LET A$="RESOURCE"
1635 RETURN
1640 LET A$="BETTER"
1645 RETURN
1650 LET A$="BUTTER"
1655 RETURN
1660 LET A$="STRAWBERRY"
1665 RETURN
1670 LET A$="WIZARD"
1675 RETURN
1680 LET A$="BOTHERSOME"
1685 RETURN
1690 LET A$="SORCERER"
1695 RETURN
1700 LET A$="ATOM"
1705 RETURN
1710 LET A$="WICKEDLY"
1715 RETURN
1720 LET A$="ENVY"
1725 RETURN
1730 LET A$="WANTON"
1735 RETURN
1740 LET A$="WANDERER"
1745 RETURN
```

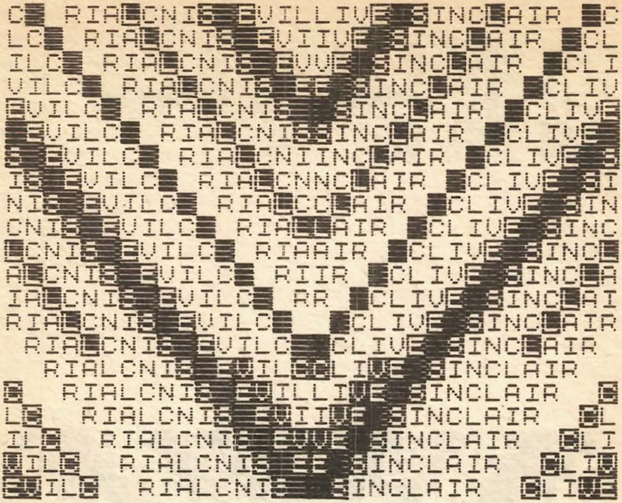
WALLPAPER

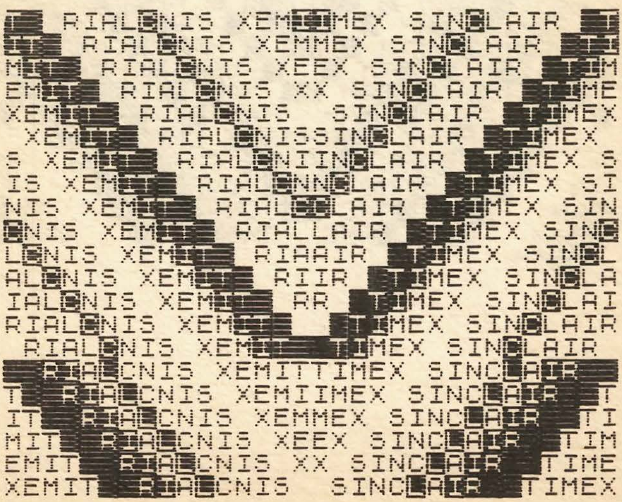
This program, written by Mark Charlton, takes your name (or any string up to 16 letters long, with spaces and/or graphics) you care to enter, and produces a continuously unfolding, and evolving, "wallpaper" pattern, as the sample run shows.

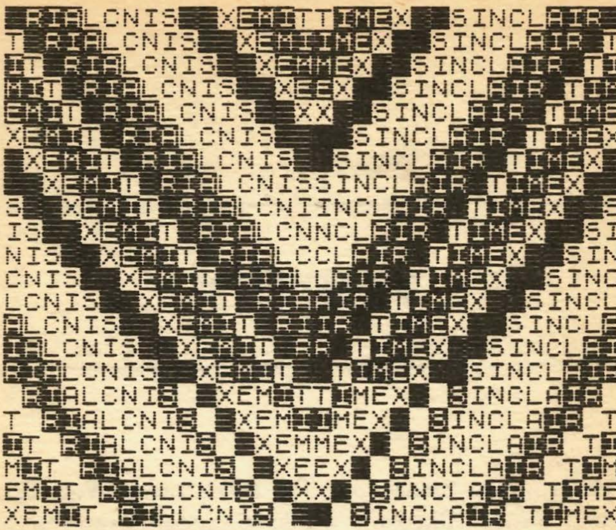
```
  10 REM NAME WALLPAPER
  20 REM (C)MARK CHARLTON 1982
  25 SCROLL
  30 PRINT "ENTER YOUR NAME"
  35 SCROLL
  40 INPUT A$
  45 LET A$=A$+" "
  46 IF LEN A$<16 THEN GOTO 45
  47 LET A$=A$( TO 16)
  50 FOR G=1 TO 16
  60 IF RND>=.5 AND CODE A$(G)<1
28 THEN LET A$(G)=CHR$ (CODE A$(
G)+128)
  70 IF RND>=.5 AND CODE A$(G)>1
27 THEN LET A$(G)=CHR$ (CODE A$(
G)-128)
  80 NEXT G
 120 FOR H=1 TO 16
 130 FOR A=-16 TO 16
 145 IF A=0 THEN GOTO 160
 150 PRINT A$(ABS A);
```

```
160 NEXT A
170 SCROLL
180 LET A$=A$(2 TO )+A$(1)
190 NEXT H
200 GOTO 50
```









POETRY

This program turns your T/S 1000 into a Walt Whitman... almost. Choosing words at random from the lines from 100 on, and spacing them out at random using lines 20 to 30, the program manages to join phrases together surprisingly well.

The program checks (line 53) to ensure that the same word is not used twice in a row, and continues to add words to a line (lines 60 and 80) until the line would overflow. At this point, it prints the line to the screen and starts constructing another one.

Once you've run this a few times, change the words from lines 100 to 215, adding words and phrases of your own choice. You'll find the "poems" are more satisfactory if the words used are related to a central topic.

```
10 REM POETRY
15 SCROLL
17 IF RND>.7 THEN GOTO 40
20 FOR J=1 TO RND*3
25 SCROLL
30 NEXT J
40 LET A$=" "
50 GOSUB 100+10*INT (RND*12)
51 LET X=LEN A$
```

```
  52 LET Y=LEN B$
  53 IF A$(X-1)=B$(Y-1) THEN GOT
O 50
  60 IF X+Y>=32 THEN GOTO 90
  80 LET A$=A$+B$
  85 GOTO 50
  90 PRINT A$
  95 RUN
 100 LET B$="DETACHED "
 105 RETURN
 110 LET B$="INITIATE "
 115 RETURN
 120 LET B$="EARLY "
 125 RETURN
 130 LET B$="ALTHOUGH "
 135 RETURN
 140 LET B$="..."
 145 RETURN
 150 LET B$="DISCIPLE "
 155 RETURN
 160 LET B$="WEEPING "
 165 RETURN
 170 LET B$="ONLY "
 175 RETURN
 180 LET B$="REACHED OUT FOR "
 185 RETURN
 190 LET B$="LONELY "
 195 RETURN
 200 LET B$="YEARNS FOR "
 205 RETURN
 210 LET B$="THEN "
 215 RETURN
```

TILE CRAZY

Ken Mahogany's game *Tile Crazy* puts you in command of a 4 × 4 grid, which holds the letters of the alphabet. You have to arrange them in alphabetical order, as follows:

A B C D
E F G H
I J K L
M N O

... with a space in the bottom right-hand corner. You move by entering a number (there is a code beside the printout) of the letter you wish to move, then the square into which you wish to move it. You will not be allowed to cheat. The program counts how many moves you've made. You should be able to do it in 40 or so moves. If you wish to change the order of the letters at the start of the game, change the contents of line 345.

```
10 REM TILE CRAZY
20 REM (C) K MAHOGANY, 1982
30 GOSUB 330
40 GOSUB 200
50 GOSUB 200
90 PRINT AT 16,3;"WHICH ONE TO
MOVE?"
```

```
 100 INPUT X
 110 IF A(X)=CODE " " THEN GOTO
100
 120 PRINT AT 16,3;"           TO
WHERE? "
 130 INPUT Y
 140 IF A(Y)<>CODE " " THEN GOTO
 130
 150 LET A(Y)=A(X)
 160 LET A(X)=CODE " "
 170 LET GO=GO+1
 180 GOTO 50
 200 REM  *** PRINT OUT ***
 210 PRINT AT 0,3;"GO NUMBER ";G
O
 220 PRINT
 225 PRINT
 230 PRINT CHR$ A(1);CHR$ A(2);C
HR$ A(3);CHR$ A(4),"  1  2  3  4
"
 240 PRINT CHR$ A(5);CHR$ A(6);C
HR$ A(7);CHR$ A(8),"  5  6  7  8
"
 250 PRINT CHR$ A(9);CHR$ A(10);
CHR$ A(11);CHR$ A(12),"  9 10 11
 12"
 260 PRINT CHR$ A(13);CHR$ A(14)
;CHR$ A(15);CHR$ A(16)," 13 14 1
5 16"
 320 RETURN
 330 REM *** INITIALIZE ***
 340 DIM A(16)
 345 LET A$="DJNBGLAEO HMCKIF"
 350 FOR B=1 TO 16
 360 LET A(B)=CODE A$(B)
 370 NEXT B
 380 LET GO=1
 410 RETURN
```

WORDSQUARE

In this program, you enter a number of words which the computer then hides on a grid, whose dimensions depend upon the length of the longest word in the list. If you find the task of trying to discover where each word is hiding too difficult, the computer will obligingly pick them out for you, in inverse letters.

THE PROGRAM

The program has been designed in modules in an attempt to make it easy to understand and modify the flow.

Lines 10 to 260 are the initialization process. The words which are to be used are stored in the string array CS. The longest word must be input first so that the size of the array can be determined. A check is made in line 170 to make sure that none of the words is too long for the array. If this is the case then the word is not accepted and a new word must be input.

Lines 200 to 260 print the wordsquare grid onto the screen.

Lines 270 to 550 are the main part of the program

and actually fit the words into the square. A two-dimensional array is first set up to store the co-ordinates finally chosen for the characters in each word (H$). The current word is assigned to variable J$ and random starting co-ordinates (X and Y) and displacements (Z and W) are chosen in lines 310 to 370.

Lines 390 to 480 single-step through the word, fitting each character into the square and storing its co-ordinates *temporarily* in the two-dimensional array K. If the word runs off the square when the co-ordinates are incremented by the displacement, or the chosen co-ordinates are already filled by an unsuitable letter from another word, the current word is started again with new X,Y,Z, and W variables. Only when the current word has been completely fitted in will its characters be entered in the final array and be printed to the screen by lines 490 to 540.

Lines 560 to 650 fill all the vacant spaces on the grid with random letters. If you do not wish to see the words as they are fitted into the grid, you can specify this at the start. The program will then only print in the words as it generates the random letter.

Lines 700 to 750 will show you the positions of the words when you get bored of looking for them by inversing them on the square when requested to do so.

There is also a visual indication of the progress made on each word as the program is running.

VARIABLES USED

i) Simple numerical variables:

- A—number of words in the list.
- D—size of the square (length of longest word plus 2)
- X—X co-ordinate
- Y—Y co-ordinate
- Z—displacement to X co-ordinate
- W—displacement to Y co-ordinate

ii) Simple string variables;
  B$—longest word
  D$—current word input
  J$—current word in square
  P$—random letter
  R$—set for secret generation of square
  Q$—set for printing of answers

iii) Numerical arrays;
  K—temporary store of co-ordinates

iv) String arrays
  C$—list of words
  H$—store for final positions for each letter

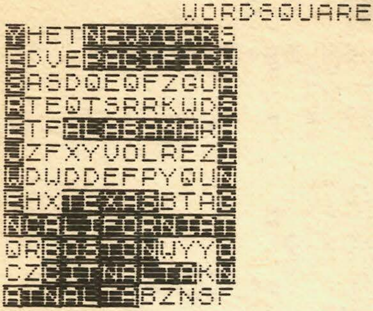
All other variables are the control variables for loops involved in input of word lists, printing to the screen, or arrays or character fitting.

The longest word in the list should have no more than 18 letters or the grid will not fit onto the screen. About 20 words of varying length can be fitted in five or ten minutes. A longer list of words can result in a very frustrating wait.

It is a good idea to enter the words in descending order of length, as this will speed up operation. The program is fascinating to watch in operation. *Wordsquare* was written by J. Elliott.

```
               WORDSQUARE
YHETNEWYORKS
EDVEPACIFICW
SASDQEQFZGUA
RTEQTSRRKWDS
ETFALABAMARH
JZFXYVOLREZI
WDWDDEFPYQUN
EHXTEXASBTAG
NCALIFORNIAT
QRBOSTONWYYO
CZCITNALTAKN
ATNALTABZNSF
```

```
FINISHED
```




```
   1 REM WORDSQUARE
   2 REM BY J ELLIOTT
  10 PRINT "IF YOU DO NOT WISH T
O SEE"
  20 PRINT "THE ANSWERS THEN ENT
ER ""N"""
  30 PRINT "NOW. OTHERWISE PRESS
 ANY KEY"
  40 LET R$=INKEY$
  50 IF R$="" THEN GOTO 40
  60 CLS
  70 PRINT AT 0,10;"WORDSQUARE"
  80 PRINT AT 19,0;"HOW MANY WOR
DS?"
  90 INPUT A
 100 PRINT AT 19,0;"ENTER LONGES
T WORD"
 110 INPUT B$
 120 DIM C$(A,LEN B$)
 130 LET C$(1)=B$
 140 FOR C=2 TO A
 150 PRINT AT 19,0;"ENTER WORD N
UMBER ";C
```

```
 160 INPUT D$
 170 IF LEN D$>LEN B$ THEN GOTO
150
 180 LET C$(C)=D$
 190 NEXT C
 199 REM NEXT LINE CONTAINS 22
          SPACES
 200 PRINT AT 19,0;"
                      "
 210 LET D=LEN B$+2
 220 FOR E=1 TO D
 230 FOR F=1 TO D
 240 PRINT AT E,F;"*"
 250 NEXT F
 260 NEXT E
 270 DIM H$(D,D)
 280 FOR Q=1 TO A
 290 LET J$=C$(Q)
 300 PRINT AT 19,0;J$
 310 LET X=INT (RND*D)+1
 320 LET Y=INT (RND*D)+1
 330 LET Z=INT (RND*3)
 340 LET W=INT (RND*3)
 350 IF Z=0 AND W=0 THEN GOTO 33
0
 360 IF Z=2 THEN LET Z=-1
 370 IF W=2 THEN LET W=-1
 380 DIM K(LEN J$,2)
 390 FOR L=1 TO LEN J$
 395 REM   SINGLE SPACE IN QUOTE
 MARKS IN NEXT LINE
 400 IF J$(L)=" " THEN GOTO 480
 410 LET X=X+Z
 420 LET Y=Y+W
 430 IF X<1 OR X>D OR Y<1 OR Y>D
 THEN GOTO 290
 435 REM   SINGLE SPACE IN QUOTE
 MARKS IN NEXT LINE
 440 IF (NOT H$(X,Y)=" ") AND (N
OT (H$(X,Y)=J$(L))) THEN GOTO 29
0
 450 LET K(L,1)=X
 460 LET K(L,2)=Y
 470 PRINT AT 19,L-1;CHR$ (CODE
J$(L)+128)
 480 NEXT L
 490 FOR M=1 TO LEN J$
 495 REM   SINGLE SPACE IN QUOTE
 MARKS IN NEXT LINE
 500 IF J$(M)=" " THEN GOTO 540
```

```
 510 LET H$(K(M,1),K(M,2))=J$(M)
 520 IF R$="N" THEN GOTO 540
 530 PRINT AT K(M,1),K(M,2);J$(M

 540 NEXT M
 550 NEXT Q
 555 REM 15 SPACES IN NEXT LINE
 560 PRINT AT 19,0;"
 "
 570 FOR N=1 TO D
 580 FOR P=1 TO D
 585 REM   SINGLE SPACE IN QUOTE
 MARKS IN NEXT LINE
 590 IF NOT H$(N,P)=" " THEN GOT
O 630
 600 LET P$=CHR$ (INT (RND*26)+3
8)
 610 PRINT AT N,P;P$
 620 GOTO 640
 640 NEXT P
 650 NEXT N
 660 PRINT AT 19,0;"FINISHED"
 670 PRINT AT 20,0;"PRESS ANY KE
Y FOR ANSWERS"
 680 LET Q$=INKEY$
 690 IF Q$="" THEN GOTO 680
 700 FOR N=1 TO D
 710 FOR P=1 TO D
 715 REM   SINGLE SPACE IN QUOTE
 MARKS IN NEXT LINE
 720 IF H$(N,P)=" " THEN GOTO 74
0
 730 PRINT AT N,P;CHR$ (CODE H$(
N,P)+128)
 740 NEXT P
 750 NEXT N
```

THE TIMEX SINCLAIR 1000™ AND 1500™ HAVE BROUGHT THE POWER OF THE COMPUTER AGE INTO OVER A MILLION HOMES.

And now hours of fun and entertainment await you at the touch of a keyboard with these easy-to-set-up programs for:

MOVING GRAPHIC GAMES LIKE *GALAXY PATROL* AND *BALLOON BUSTER*

DRIVING GAMES LIKE *ALLEY DRIVER* AND *SQUEEZER IV*

BOARD GAMES AND SIMULATIONS LIKE *ETCHASKETCH* AND *MUSIC*

WORD AND LETTER GAMES LIKE *ANAGRAMS* AND *TILE CRAZY*

Whether you like thought games or action games, roaming the far reaches of space or the ocean depths, drawing pictures or solving puzzles, you'll find something to challenge your game-playing skills in—

51 GAME PROGRAMS FOR THE TIMEX SINCLAIR 1000™ AND 1500™

0 71162 00250 12598

ISBN 0-451-12598-3